

BEST AVAILABLE COPY

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

CORRECTED VERSION

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
1 November 2001 (01.11.2001)

PCT

(10) International Publication Number
WO 01/82533 A2

(51) International Patent Classification: **H04L 12/46**

Neck, NJ 07722-1348 (US). **MACEY, Christopher** [US/US]: 54 Elm Place, Red Bank, NJ 07701-1928 (US). **HARWOOD, Jonathan** [US/US]: 8 Post Road, Rumson, NJ 07760 (US). **FRANCUS, Jerold** [US/US]: 1017 Old Chester Road, Fair Hills, NJ 07931 (US). **TUOMENOKSA, Mark** [US/US]: 464 South Border Road, Winchester, MA 01890 (US). **BENDINELLI, Samuel** [US/US]: 36 Harvard Circle, Princeton, NJ 08540 (US).

(21) International Application Number: **PCT/US01/08970**

(22) International Filing Date: 22 March 2001 (22.03.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/196,297 12 April 2000 (12.04.2000) US

(74) Agents: **GARRETT, Arthur, S. et al.**: Finnegan, Henderson, Farabow, Garrett & Dunner, L.L.P., 1300 I Street, N.W., Washington, DC 20005-3315 (US).

(71) Applicant (for all designated States except US): **OPEN-REACH.COM** [US/US]: Suite 104, 760 State Route 18, East Brunswick, NJ 08816-47907 (US).

(81) Designated States (national): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

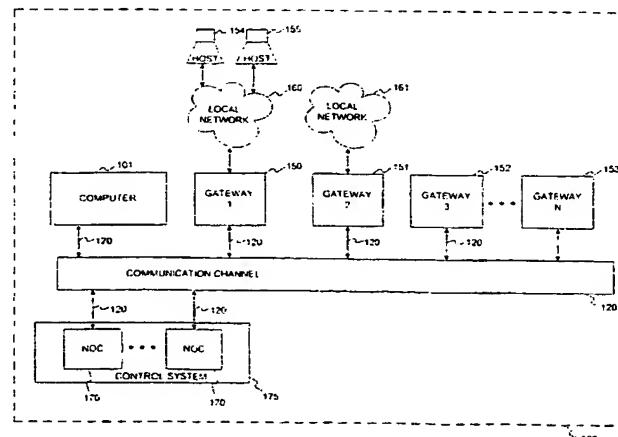
(72) Inventors; and

(75) Inventors/Applicants (for US only): **KEANE, John** [US/US]: 37 Memorial Parkway, Metuchen, NJ 08840-2138 (US). **SHIMAMOTO, Brion** [US/US]: 343 Riverside Avenue, Riverside, CT 06878-2123 (US). **HERRICK, Michael** [US/US]: 20 Rimwood Lane, Colts

(84) Designated States (regional): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian

[Continued on next page]

(54) Title: METHOD AND SYSTEM FOR MANAGING AND CONFIGURING VIRTUAL PRIVATE NETWORKS



(57) Abstract: Methods and systems are provided for enabling a network between a first and a second processor using at least one additional processor separate from the first and second processors. In one embodiment, the at least one additional processor receives information indicating a consent on behalf of the first processor to enabling a tunnel between the first processor and the second processor and receives information indicating a consent on behalf of the second processor to enabling a tunnel between the second processor and the first processor. The at least one additional processor determines a first virtual address for the first processor and a second virtual address for the second processor such that the first and second virtual addresses uniquely identify the first and second processors, respectively, and are routable through the network. The at least one additional processor provides to each of the first and second processors the first and second virtual addresses to enable one or more tunnels between the first and the second processors.

WO 01/82533 A2



patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM). European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR). OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:

- without international search report and to be republished upon receipt of that report

(15) Information about Correction:

see PCT Gazette No. 52/2001 of 27 December 2001, Section II

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(48) Date of publication of this corrected version:

27 December 2001

(19) World Intellectual Property Organization
International Bureau(43) International Publication Date
1 November 2001 (01.11.2001)

PCT

(10) International Publication Number
WO 01/82533 A2(51) International Patent Classification⁷: H04L 12/46

TUOMENOKSA, Mark [US/US]; 464 South Border Road, Winchester, MA 01890 (US). BENDINELLI, Samuel [US/US]; 36 Harvard Circle, Princeton, NJ 08540 (US).

(21) International Application Number: PCT/US01/08970

(74) Agents: GARRETT, Arthur, S. et al.; Finnegan, Henderson, Farabow, Garrett & Dunner. L.L.P., 1300 I Street, N.W., Washington, DC 20005-3315 (US).

(22) International Filing Date: 22 March 2001 (22.03.2001)

(81) Designated States (national): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(25) Filing Language: English

(84) Designated States (regional): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

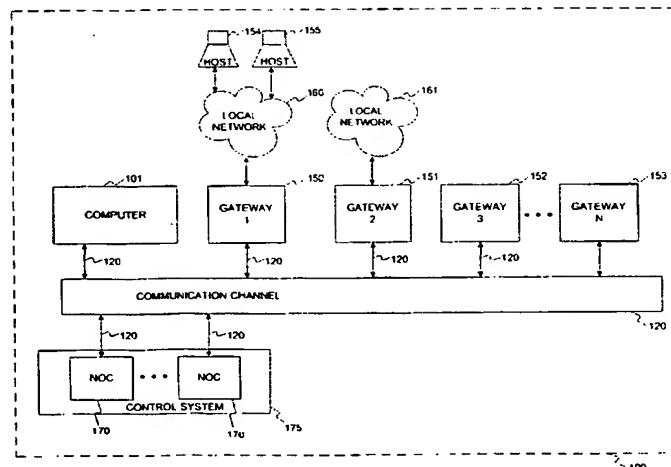
(26) Publication Language: English

Published:

— without international search report and to be republished upon receipt of that report

[Continued on next page]

(54) Title: METHOD AND SYSTEM FOR MANAGING AND CONFIGURING VIRTUAL PRIVATE NETWORKS



(57) **Abstract:** Methods and systems are provided for enabling a network between a first and a second processor using at least one additional processor separate from the first and second processors. In one embodiment, the at least one additional processor receives information indicating a consent on behalf of the first processor to enabling a tunnel between the first processor and the second processor and receives information indicating a consent on behalf of the second processor to enabling a tunnel between the second processor and the first processor. The at least one additional processor determines a first virtual address for the first processor and a second virtual address for the second processor such that the first and second virtual addresses uniquely identify the first and second processors, respectively, and are routable through the network. The at least one additional processor provides to each of the first and second processors the first and second virtual addresses to enable one or more tunnels between the first and the second processors.

WO 01/82533 A2



For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

METHOD AND SYSTEM FOR MANAGING AND CONFIGURING VIRTUAL PRIVATE NETWORKS

CROSS REFERENCE TO RELATED APPLICATION

This application claims the benefit of U.S. Provisional Patent Application No. 60/196,297, entitled "A METHOD AND SYSTEM FOR MANAGING VIRTUAL PRIVATE NETWORKS," filed on April 12, 2000, the disclosure of which is expressly incorporated herein by reference in its entirety.

DESCRIPTION OF THE INVENTION

Field of the Invention

The present invention relates to systems and methods for controlling networks, and in particular, to systems and methods for implementing virtual private networks.

Background of the Invention

Wide area networks allow users to access company files and computer programs, regardless of where users are geographically located. Until recently, building wide area networks remained the province of only the largest corporations or companies with enough technical skill and financial resources. Organizations have used a range of approaches to building wide area networks to connect remote offices, partners, or employees. These "traditional" approaches to connectivity include, for example, point-to-point leased lines, packet switched networks, and dedicated virtual private networks (VPNs).

Point-to-point leased lines are physical networks requiring the engineering of separate links between sites that need to communicate with each other. Point-to-point leased lines can take from 30 to 90 days to install and are costly.

A packet switched network using frame relay is a traditional alternative to point-to-point leased lines that offers reduced costs and increased flexibility. Like the point-to-point solutions, the initial installation of a frame relay network takes a long time. For example, additional access circuits may usually take two to three weeks for installation and the service is fairly costly.

A more-recently introduced service offered by some network service providers is a dedicated virtual private network. This routed service eliminates the complexity and costs associated with the engineering of connections between dedicated locations, but requires the network service provider to manage security as the network is shared with other customers. A virtual private network is "virtual" because it uses a shared or a base network, such as the Internet as its backbone as opposed to a completely private network with dedicated lines. It is also "private" since the information that is exchanged between the users may be encrypted or encoded to provide privacy. Prior to the present invention, virtual private networks, dedicated point-to-point lines, and packet switched networks shared drawbacks of being cumbersome and costly.

Although traditional virtual private networks offer low access costs, they often entail high set-up, maintenance, and management costs. Based on a number of factors, a shared network such as the Internet has evolved as the preferred backbone for connecting and internetworking multiple locations, partners, and employees. Also, the Internet offers the advantages of being ubiquitous, (available almost everywhere – small towns, large cities, around the world), offering an enormous capacity, and increasing cost-effectiveness, with fast, new access methods, such as DSL and cable modems.

With the advent and ubiquity of the Internet, virtual private networks have emerged as a way to build a private communication network over a shared public or private infrastructure or a base network. Virtual private networks provide secure private connections over the Internet by enabling authentication of users and locations, delivering secure and private "tunnels" between users or locations, and encrypting user communications.

Today, most virtual private networks are Internet Protocol (IP) based and are established over the Internet. They fall into two categories, namely hardware-based and software-based virtual private networks. Hardware-based virtual private networks require proprietary hardware platforms and claim to provide high price/performance ratios and potentially increased security through specialized functions. Network manufacturers are building

some virtual private network capabilities into routers and other networking equipment.

Software-based virtual private networks have emerged as another alternative to hardware-based virtual private networks. Vendors are already adding virtual private network functionality, such as tunneling and encryption to their firewall solutions.

Although use of a base network, such as the Internet as a backbone for wide area networks may be less expensive and more flexible than traditional solutions, the associated costs and complexity of using virtual private networks has been prohibitive. As a result, most companies have been reluctant to link remote locations over the Internet using virtual private networks.

Building wide area virtual private networks over the Internet has been difficult because most robust solutions have required esoteric networking and security technologies. Merely deciding what type of virtual private network and what levels of security or encryption are required can be confusing to many information technology (IT) personnel and non-IT personnel. Beyond the complex purchase decisions, the installation and ongoing maintenance of such systems can be time-consuming, especially if the number of remote locations changes frequently. In addition, many companies have found that rolling out traditional virtual private network products requires significant logistical planning to make sure that the right hardware and software is available at all the remote locations. Initial configuration of these remote sites is often time consuming enough, without factoring in the effort required to get a remote site back on line if a location fails (especially if no skilled IT resources are available at the remote site).

Many organizations have been reluctant to establish Internet-based wide area virtual private networks also because of the increasing number of Internet security threats, such as hackers and corporate espionage. Further, virtual private networks and Internet-based connectivity solutions continue to remain prohibitively expensive. Even prepackaged virtual private network solutions require expensive networking personnel to configure, install, and

manage such networks. For example, enterprise level firewall and virtual private network solutions may take up to a week to configure. In addition, the installation often requires support at the remote locations, dictating either extensive travel requirements for home office personnel or the hiring and training of remote IT support staff.

Many software-based virtual private network solutions also require the purchase of specialized and costly hardware. Moreover, although virtual private networks can save considerable amounts of money over frame relay or leased line networks, associated IT support costs often erase the savings. For example, setting up a virtual private network may necessitate hiring full-time IT professional to set up and administer the network.

As explained above, the installation and maintenance of a secure virtual private network over the Internet have been too complex, requiring financial investment in hardware, software, personnel, and/or time. To provide encryption and authentication on a virtual private network, each user must perform a variety of tasks including, for example, using an encryption algorithm that is compatible with the virtual private network; using an authentication technique that is compatible with the virtual private network; coordinating various security protocols with other users (e.g., coordinating a public key exchange) of the virtual private network; coordinating the establishment of tunnels with other users of the virtual private network; selecting and manually configuring the encryption path through the communication path; and/or recovering the virtual private network after a failure. Accordingly, the burdens of installing and administering virtual private networks are significant.

SUMMARY OF A FEW ASPECTS THE INVENTION

To address the above and other limitations of the prior art, methods and systems are provided that easily and effectively leverage the power of a shared or a base network, such as the Internet for private connectivity without

-5-

the complexity, cost, or time associated with setting up traditional virtual private networks. Rather than requiring specialized hardware, such methods and systems are capable of being self-configured on nonproprietary hardware, such as a standard personal computer (PC), to quickly establish one or more virtual private networks over a local or wide geographical area. Configuration may be achieved by pointing-and-clicking, making it feasible for users to build secure virtual private networks.

Methods and systems consistent with the present invention enable one or more networks between a first processor and a second processor using at least one additional processor separate from the first and second processors. The additional processor receives information indicating consent on behalf of the first processor to enabling a tunnel between the first processor and the second processor and information indicating consent on behalf of the second processor to enabling a tunnel between the second processor and the first processor. The additional processor determines a first virtual address for the first processor and a second virtual address for the second processor such that the first and second virtual addresses uniquely identify the first and second processors, respectively, and are routable through the network. The additional processor provides to each of the first and second processors the first and second virtual addresses to enable one or more tunnels between the first and the second processors, thus enabling one or more networks between the first and second processors.

Furthermore, methods and systems consistent with the present invention may provide program code that configures a processor, such as the

-6-

first processor into a gateway capable of being enabled by the additional processor to establish one or more tunnels to another processor, such as the second processor through a communication channel.

It is to be understood that both the foregoing general description and the following detailed description are exemplary and explanatory only and are not restrictive of the invention, as described. Further features and/or variations may be provided in addition to those set forth herein. For example, the present invention may be directed to various combinations and subcombinations of the disclosed features and/or combinations and subcombinations of several further features disclosed below in the detailed description.

The accompanying drawings, which are incorporated in and constitute a part of this specification, illustrate several embodiments of the invention and together with the description, serve to explain the principles of the invention.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a general block diagram of a first exemplary network in accordance with methods and systems consistent with the present invention;

FIG. 2 is a general block diagram of an exemplary processor in which systems and methods consistent with the present invention may be implemented;

FIG. 3 is an exemplary flow chart for initially registering with a control system in accordance with methods and systems consistent with the present invention;

FIG. 4 is a general block diagram of a second exemplary network in accordance with methods and systems consistent with the present invention;

FIG. 5 is an exemplary flow chart for establishing a network in accordance with methods and systems consistent with the present invention;

FIG. 6A is a general block diagram of a third exemplary network in accordance with methods and systems consistent with the present invention;

FIG. 6B shows virtual IP addresses for a network in accordance with methods and systems consistent with the present invention;

FIG. 7 is an exemplary flow chart for providing information to a Network Operations Center (NOC) in accordance with methods and systems consistent with the present invention;

FIG. 8 is an exemplary flow chart for defining a gateway in accordance with methods and systems consistent with the present invention;

FIG. 9 is an exemplary flow chart for creating a program code for configuring a processor as a gateway in accordance with methods and systems consistent with the present invention;

FIG. 10 is an exemplary flow chart for configuring a processor as a gateway in accordance with methods and systems consistent with the present invention;

FIG. 11 illustrates exemplary partner lists in accordance with methods and systems consistent with the present invention;

FIG. 12 is an exemplary screen for adding a gateway to the virtual private network in accordance with methods and systems consistent with the present invention;

FIG. 13 is an exemplary flow chart for establishing a tunnel in accordance with methods and systems consistent with the present invention;

FIG. 14 is a general block diagram of a tunnel between two gateways in accordance with methods and systems consistent with the present invention;

FIG. 15 is a general block diagram of two gateways, each not accessible behind a firewall, in accordance with methods and systems consistent with the present invention;

FIG. 16A is a general block diagram of a tunnel between a gateway and a network operations center in accordance with methods and systems consistent with the present invention;

FIG. 16B is a general block diagram of a tunnel between a network operations center and a gateway that includes a client computer in accordance with methods and systems consistent with the present invention;

FIG. 17 is an exemplary flow chart for performing the protocol associated with a connection from a gateway to a network operations center in accordance with methods and systems consistent with the present invention;

FIG. 18 is a general block diagram of an alternative exemplary network in accordance with methods and systems consistent with the present invention;

FIG. 19 is an exemplary flow chart for detecting an address change in a network in accordance with methods and systems consistent with the present invention;

FIG. 20 is an exemplary flow chart for resolving address conflicts in a local network in accordance with methods and systems consistent with the present invention; and

FIG. 21 is a general block diagram of another exemplary network in accordance with methods and systems consistent with the present invention.

DETAILED DESCRIPTION

Reference will now be made in detail to the exemplary embodiments of the invention, examples of which are illustrated in the accompanying drawings. Wherever possible, the same reference numbers will be used throughout the drawings to refer to the same or like parts.

In accordance with an embodiment of the present invention, a prospective user or customer may contact a mediation point or a control system, such as a network operations center via a base network, such as the Internet, and indicate a desire to establish one or more virtual private networks. After answering a series of questions posed by the network operations center, the user receives program code and information for loading onto one or more processors, such as personal computers. The program code and information may be in the form of a disk, such as an optical disk or floppy disk, downloaded over the Internet and onto a disk, or installed directly

over the Internet on to a computer. The program code may be distributed to other computers at other desired sites user sites as well. Alternatively, the program code and information may be preinstalled on a computer and delivered to the user.

The user then runs or boots a computer with the provided code and information. When the computer is booted, it thereafter communicates with the network operations center over the Internet to receive further information such that the computer is configured as a gateway or a computer capable of participating in one or more virtual private networks enabled by the network operations center over a base network, such as the Internet. The provided code and information may also be loaded on other computers such that the computer is configured as a gateway.

After configuration is completed and based on the user's request, the network operations center may enable over the Internet one or more virtual private networks between the gateway and other gateways configured through the network operations center. At the consent of the user, the virtual private networks may be periodically reconfigured to add additional gateways at, for example, geographically dispersed sites or to provide full or limited access to the networks via other gateways.

Consequently, the user may configure one or more gateways using a computer, such as a personal computer, without investing in costly proprietary hardware or setting up a typically costly network administration department. Because the gateway as configured is not dependent on a particular piece of hardware, flexible virtual private networks may be inexpensively established between remote locations.

Accordingly, the user may choose and change its Internet service providers (ISPs), network equipment, and access types (T1, cable modem, DSL, etc.) and then access the network operations center through the Internet to update configuration information that may have resulted from such a change. Furthermore, to participate in a virtual private network, a user need not require other users to use specific network gear or sign-up with specific ISPs. Instead, the user may direct other users to the network operations

-10-

center to receive program code and information to configure one or more gateways capable of participating in one or more virtual private networks.

The user may quickly bring up new gateways in minutes rather than weeks or months. As explained above, the user may install the program code, log onto a network operations center with any web browser, and connect to London, New York and Boston in minutes. Unlike traditional virtual private network services requiring 30 to 90 days for installation of a new Internet connection, the gateways may be configured to be compatible with the user's existing Internet connections. The user may even start with a dial-up or ISDN connection and later replace it with a faster DSL, cable, or T1 connection without affecting service. Additionally, unlike traditional network equipment requiring expensive overnight shipping, the gateway program code may be downloaded almost anywhere in the world or may be distributed on a storage device, such as an optical disk or a floppy disk.

In another embodiment, two or more users may register with a controller or network operations center using a web browser. The network operations center may prompt them to provide basic identifying information, such as the Internet Protocol (IP) addresses of their computers. The network operations center may then generate a program code and configuration information and provide them to each user. After the users install the program code and configuration information on their respective computers, the respective computers establish communication with the network operations center to obtain additional configuration information for configuring themselves as gateways. After configuration is completed, one or more of the computers communicates its consent to the network operations center for establishing a tunnel to the other computer. Each computer may communicate its consent mutually and/or independently of the other computer.

If both gateways consent, the network operations center then proceeds to enable a tunnel between the user computers. The network operations center may enable the tunnel by providing sufficient information to each computer over the Internet such that the computer may establish the tunnel

with the provided information. Once the tunnel is enabled, the computers may establish the tunnel and then use the tunnel to exchange information in a secure and trusted manner. At any time, each computer may withdraw its consent and terminate the tunnel. Furthermore, other computers configured through the network operations center may also join the virtual private network.

Consequently, the tasks of installing a gateway, establishing a virtual private network, and joining a virtual private network are simplified from the perspective of the users, even when establishing a temporary virtual private network for a short term project or a short term financial transaction (e.g., a purchase or sale).

As such, the described methods and systems may be for various applications, such as, for example, enabling the establishment of virtual private networks without costly hardware and software outlays; providing virtual private networks to businesses that sell products to customers over the Internet; providing virtual private networks to users of a corporate Intranet that seek to share information with outside users in a secure manner; and providing virtual private networks to users of the Internet in general. In such applications, the users may communicate with the virtual private networks by registering over the Internet with a control system, such as a network operations center; installing a program code; and indicating a consent to participate in a virtual private network. As a result, managing virtual private networks is simplified since users are not required to, for example, coordinate selection of encryption algorithms and/or authentication techniques; monitor and/or control tunnels of virtual private networks; and/or recover virtual private networks from failures.

From a business perspective, the user may be charged a periodic fee based on the number of gateways configured by the user through the network operations center. Alternatively, charges might also be assessed based on one or more of the following: the volume of information transported on the virtual private networks, the number of tunnels, or the usage time.

Before embarking on an element-by-element description of various preferred embodiments, the following terms are described. A gateway refers to any processor through which access is provided to a network. For example, a gateway may provide hosts or computers in a local area network or in a wide area network access to another network. A processor may include, for example, a personal computer, router, bridge, server, or any other network device. An encrypted information flow includes a flow of information that is encrypted. An example of an encrypted information flow is a tunnel, such as an encrypted tunnel. A tunnel may be established, for example, when two gateways open a channel of communication through a base network, such as the Internet. A tunnel may be enabled, for example, when a gateway is provided with authorization and/or sufficient information that may be used by the gateway to establish a tunnel with another gateway.

FIG. 1 shows a general block diagram of a network 100, in accordance with an embodiment of the present invention. The network 100 may include a control system 175 with one or more network operations centers 170, a communication channel 120, one or more gateways 150-153, one or more local networks 160, 161, one or more hosts 154, 155, and a computer 101. The communication channel 120 may include a shared or base network, such as the Internet to facilitate communication and exchanges between the various entities depicted in the network 100 of FIG. 1.

In accordance with an embodiment of the present invention, a first gateway, such as gateway 150 may establish through communication channel 120 a first encrypted information flow to the control system 175. This first encrypted information flow may permit the control system 175 to exchange control information through the communication channel 120 with the first gateway 150. Further, a second gateway, such as gateway 151 may establish through communication channel 120 a second encrypted information flow to the control system 175. This second encrypted information flow may also permit the control system 175 to exchange with the second gateway 151 control information through the communication channel 120. Since both of

these information flows may be encrypted, the encrypted information flow may provide privacy.

The control system 175 may also enable a third encrypted information flow through the communication channel 120 between the first gateway 150 and the second gateway 151. The control system 175 may enable the third encrypted information flow after the first gateway 150 and the second gateway 151 consent to enabling the third encrypted information flow.

The consent communicated to the control system 175 may be mutual in that the first gateway 150 and the second gateway 151 each consents to enabling of the third tunnel. Moreover, the consent may be independent in that the first gateway 150 and the second gateway 151 independently consent to the establishment of the third tunnel without regard to whether the other gateway consents. A gateway may communicate its consent by identifying the names and/or addresses of the other gateways. For example, in an embodiment, a gateway may identify its consent to enabling a tunnel with another gateway by simply providing the name of the other gateway to the control system 175. If the control system 175 determines that the consent is mutual (i.e., that the other gateway also consents to enabling the tunnel), the control system 175 places the other gateway on a list (hereinbelow referred to as a partner list) that will be provided to the gateway. Likewise, the control system places the gateway on the partner list for the other gateway. That is, the control system 175 places each gateway on the partner list of the other gateway and provides the respective partner lists to each gateway. Accordingly, the partner list reflects the mutual desire of each gateway to enable a tunnel.

For example, referring to FIG. 1, a user using host computer 155 may use a web browser to access the control system 175 through the tunnel between gateway 150 and the control system 175. The control system 175 may then provide the user with the names of other gateways that gateway 150 may establish a tunnel with (e.g., the names for gateways 151-153). The user then may select one or more names corresponding to the other gateways that gateway 150 consents to enabling a tunnel with. The user may

then submit the names of the selected gateways to the control system 175, which determines if there is mutual consent for each of the selected gateways. That is, the control system 175 determines for each of the selected gateways whether or not the selected gateway also consents to enabling a tunnel with gateway 150. If there is mutual consent, each of the selected gateways that also consents is added to the partner list for gateway 150, and gateway 150 is also added to the partner list for each of the selected gateways. These partner lists may then be forwarded by the control system 175 to gateway 150 and each of the selected gateways.

Accordingly, when the control system 175 determines that the first gateway 150 and the second gateway mutually consent to the third tunnel, the control system may then provide to the first and second gateways through the first and second tunnels, respectively, sufficient information to enable the third tunnel. The third tunnel may be enabled, for example, when the first and second gateways are provided sufficient information allowing them to establish this third tunnel through the communication channel 120. In one embodiment, the sufficient information includes the partner list for the first gateway and the partner list for the second gateway. Moreover, for each gateway listed on the partner list, the partner list may include, for example, a virtual IP address, a real IP address, and/or other information describing each gateway. After the third tunnel is enabled, the first and second gateways 150, 151 may establish the third tunnel through the communication channel 120. This third tunnel may provide privacy as to the exchanged information and may also be authenticated using an Internet Protocol Security (IPSec) compliant authentication technique, such as MD-5 hashing. Also, the encryption used for the encrypted information flow may be a weak encryption or encoding algorithm that provides minimal privacy or may be a strong encryption scheme that essentially guarantees privacy.

An encrypted information flow, such as a tunnel may be established through communication channel 120 by, for example, encapsulating a protocol within another protocol. For example, a tunnel may be encrypted when an Internet Protocol packet encapsulates an encryption protocol.

-15-

Examples of encryption protocols may include RSA, Digital Encryption Standard (DES), and Triple DES (3DES). For example, an encrypted tunnel may be established using Internet Protocol (IP) packets such that the payload of each packet is encrypted but the address of each packet is unencrypted (i.e., clear-text). As a result, the encrypted payload may be encapsulated by a clear text IP address, forming a virtual tunnel through a base network, such as the communication channel 120. Other encrypted tunnels may be established through the communication channel 120 with other gateways, such as gateways 152 and 153. These virtual tunnels established through the base network and enabled by the control system 175 may also form a virtual network. If a virtual network enabled by the control system 175 uses some type of encoding or encryption for privacy, the virtual network may also be referred to as a virtual private network.

In the embodiment of FIG. 1, the computer 101 may include, for example, a personal computer and/or a workstation that include a web browser, such as the Netscape Navigator developed by Netscape or the Internet Explorer developed by Microsoft. The computer 101 may connect to the control system 175 through the communication channel 120 using the web browser. Once the computer 101 connects to the control system 175, a user may register one or more gateways with the control system 175 and define an initial configuration for one or more of the gateways 150-153 desiring to participate in one or more virtual private networks.

After the initial configuration of the gateways 150-153 is defined, the control system 175 may create a disk image that includes program code and information for configuring the gateways 151-153. The disk image may include, for example, a copy of the program code required to configure a personal computer as a gateway. Alternatively, the control system 175 may install through the communication channel 120 a bootable program on the gateways 151-153. After executing the bootable program on a computer, the bootable program may retrieve additional program code and configuration information from the control system 175 or other secured site to configure the computer as a gateway. Moreover, the program code may be loaded onto the

-16-

gateways 150-153 using a single disk (not shown) and/or downloaded through the communication channel 120. Once the program code is installed, the gateways 150-153 may be capable of being enabled by the control system 175 and participating in one or more virtual networks or virtual private networks through the communication channel 120.

The disk image may include program code for one or more of the following: program code for IPSec; program code for communications between network operations center 170 and gateways 151-153; the Linux Operating System (OS) including kernel and device drivers; the configuration of the IP stack such as a Dynamic Host Configuration Protocol (DHCP) client and a DHCP Server; program code for routing packets through one or more tunnels established between gateways 151-153; access control information for limiting the functions performed through one or more tunnels established between gateways 151-153; program code for the SOCKS Proxy code; program code for a web browser; and any other software that may be installed based on the user's configuration. In addition, the LINUX operating system may be a "hardened" version of Linux to improve the security of the operating system. When each of the gateways 150-153 loads the disk image, each gateway may execute the program code contained in the disk image. As each of the gateways 151-153 performs the steps contained in the program code, each may connect to the control system 175 and establish an encrypted information flow to the control system 175.

The control system 175 may also enable an encrypted information flow between at least two gateways, permitting them to exchange information or traffic in a private manner. Further, the control system 175 may control and/or monitor the encrypted information flows in the network 100 by exchanging control and/or monitoring information with the gateways over the encrypted information flow.

Referring to FIG. 1, the control system 175 may include one or more network operation centers 170. Each of the network operation centers 170 may be located at the same location or may be distributed along the communication channel 120 connecting the distributed network operation

-17-

centers 170. If the network operations centers 170 are distributed, they may also use one or more gateways configured as described above to provide privacy and/or authentication. The control system 175 and the network operation centers 170 may be implemented with at least one processor including, for example, one or more of the following components: a central processing unit, a co-processor, a memory, a storage device, an input device, an output device, a network interface, a display, and/or other processing devices and systems.

The gateways 150-153 may each include, for example, one or more of the following processors: a computer, a server, a router, a switch, a portable device such as a cell phone or a personal digital assistant, or any other communication device capable of performing the functions of the gateway in accordance with the present invention. A gateway may participate as a stand-alone node or computer interfacing the communication channel 120 (see, e.g., the gateways 152 and 153) and/or as a gateway interfacing a local network (see, e.g., the gateways 150 and 151). In a stand-alone configuration, for example, the gateway 153 may permit a user to participate in one or more virtual private networks established over communication channel 120. In a local network configuration, for example, the gateway 150 may interface the local network 100 to permit one or more users, such as hosts 154 and 155 to participate in one or more virtual private networks established over communication channel 120. Furthermore, in the local network configuration, the gateway may resolve address conflicts that may exist with the local area network 160 and other networks such as local area network 161.

The host computers 154 and 155 may each include a processor, such as a computer 200 shown in FIG. 2. The computer 200 may include an input module 205, a central processing unit (CPU) 220, a storage module 250, and an output module 230. The output module 230 may include a display 235, a printer 236, and a network interface 238. One of ordinary skill in the art will recognize that each host computer 154 and 155 may also function as a

gateway in accordance with the present invention. Although FIG. 2 shows a computer 200, other devices, such as printers, personal digital assistants, wireless devices, and mobile phones, may function as a host computer and participate in one or more virtual private networks established over communication channel 120.

The input module 205 of Fig. 2 may be implemented with a variety of devices to receive a user's input and/or provide the input to the CPU 220. Some of these devices (not shown) may include, for example, a network interface module, a modem, a keyboard, a mouse, and an input storage device.

Although Fig. 2 illustrates only a single CPU 220, computer 200 may alternatively include a set of CPU. The CPU 220 may also include, for example, one or more of the following: a co-processor, memory, registers, and other processing devices and systems as appropriate.

The storage module 250 may be embodied with a variety of components or subsystems including, for example, a hard drive, an optical drive, a general-purpose storage device, a removable storage device, and/or other devices capable of storing. Further, although storage module 250 is illustrated in Fig. 2 as being separate or independent from CPU 220, the storage module and CPU 220 may be implemented as part of a single platform or system.

Referring again to FIG. 1, the communication channel 120 may facilitate communication between the various entities depicted in the network 100. The communication channel may include, for example, a telephony-based network, a local area network (LAN), a wide area network (WAN), a

dedicated Intranet, the Internet, and/or a wireless network. Further, any suitable combination of wired and/or wireless components and systems may be incorporated into the communication channel 120. Any suitable combination of point-to-point communications or network communications may also be incorporated into communication channel 120 to facilitate communication between the entities illustrated in FIG. 1. Moreover, although local networks 160, 161 are shown as being separate from the communication channel 120, the local network 160, 161 may be implemented in the same manner as the communication channel 120 or include one or more of the features of the communication channel 120.

In one embodiment, a user may serve as an administrator and may register at least one of the gateways 150-153 through control system 175 and/or establish one or more virtual private networks over communication channel 120. The user may use an Internet browser on computer 101 to contact the control system 175, to register at least one of the gateways 150-153, and/or establish one or more virtual private networks over communication channel 120. Moreover, although the computer 101 is shown as a stand-alone entity in the embodiment of FIG. 1, the computer 101 may alternatively be co-located with one or more of the gateways 150-153, the control system 170, and/or the communication channel 120.

Furthermore, the user may register with the control system 175 and provide basic information, such as the number of gateways participating in the virtual private network and billing information. Once registered, the user may receive code generated by the control system 175. The user may then reboot a computer with the received code to configure the computer as a gateway. That is, the administrator may install the code on any computer that the administrator desires to configure as a gateway including the computer serving as the computer 101. The configured gateway may then establish a tunnel to another gateway (i.e., similarly configured by the control system 175) after the control system 175 determines that each gateway mutually consents to enabling the tunnel and provides each gateway with sufficient information to enable the tunnel.

-20-

FIG. 3 shows an exemplary flowchart for initially registering one or more gateways with the control system 175. Referring to FIGS. 1 and 3, the user may register at least one of the gateways 150-153 with the control system 175 (step 310) and define a configuration for the registered gateways 150-153 (step 320). In one embodiment, the user may contact the control system 175 through the Internet using a web browser to specify a particular configuration for a gateway. This specified configuration information may include a name for the gateway and a name for the virtual private network. This name for the virtual private network will hereinafter be referred to as the virtual private network's domain name.

The control system 175 may use the specified configuration to assemble code and information, such as program code and textual information (e.g., Extensible Markup Language also referred to as "XML"), in the form of a disk image (step 330). This disk image may include all the program code and information needed to configure gateways 150-153 for establishing one or more virtual private networks established over communication channel 120. The disk image may then be provided to the user and installed on a processor, such as a personal computer or a general-purpose computer (step 340). When the processor reboots, it uses the information provided in the disk image to configure itself as a gateway capable of establishing secure tunnels to the control system 175. The disk image may be sized to fit on a single storage medium, such as a floppy disk or optical disk. Moreover, the disk may be distributed through alternative channels of distribution, such as direct mail, unsolicited mail, over-the-counter retail, or may be distributed with other hardware and software provided by a vendor. Alternatively, the disk image may be downloaded from the control system onto a storage medium or may be stored at the control system 175 for later transfer to the gateways 150-153. Accordingly, a commercial-off-the-shelf computer may be configured as a gateway capable of participating in one or more virtual private networks established over communication channel 120.

The control system 175 may perform various functions including, for example, enabling tunnels between two or more gateways in network 100; assembling and/or configuring a user's computer as a gateway; negotiating an authentication technique; determining one or more partner lists for the gateways 150-153; administering the configuration of virtual private networks established over communication channel 120; providing virtual Internet Protocol (IP) addresses to each gateway; monitoring and/or controlling the established virtual private networks; enabling the establishment of tunnels between two or more gateways in the network 100; enabling the establishment of tunnels with gateways not accessible behind firewalls; and/or recovering the established virtual private networks after a failure. The control system 175 may exchange control information with each of the gateways 150-153 through a tunnel established through the communication channel 120. Moreover, each pair of the gateways 150-153 may exchange information through one or more tunnels established between the gateways.

FIG. 4 shows an exemplary virtual private network 400 established over the communication channel 120. This exemplary network 400 will be used to illustrate how such a network is enabled. The network 400 includes a first gateway 450, a second gateway 451, a computer 401, a first tunnel 425, a second tunnel 426, a third tunnel 423, and the control system 175. The first tunnel 425, the second tunnel 426, and the third tunnel 423 may be established through the communication channel 120. Moreover, gateway 450 and gateway 451 may each participate as a stand-alone node in the virtual private network 400 or as a node interfacing a local network, such as local network 160 shown in FIG. 1.

The virtual private network 400 may be established after each of the gateways 450, 451 establishes a tunnel (e.g., the first tunnel 425 and the second tunnel 426) to the control system 175; after the first gateway 450 and the second gateway 451 each communicate to the control system 175 a consent to enable the third tunnel 423 between the first gateway 450 and the second gateway 451; after the control system 175 provides to the first gateway and the second gateway sufficient information to enable the third

-22-

tunnel 423; and after the first gateway 450 and the second gateway 451 establish the third tunnel 423. With the third tunnel established, the first gateway 450 and the second gateway 451 may communicate in a private and/or trusted manner. Although FIG. 4 only shows two gateways, additional gateways (not shown) may also join the virtual private network 400. Accordingly, the task of configuring gateways that are capable of participating in a virtual private network is significantly simplified.

A user desiring to configure the virtual private network 400 may simply register one or more gateways and administer the network through the control system 175. The tasks performed by the user may thus be simplified to, for example, initially registering with the control system, rebooting one or more computers with software provided by the control system to configure the computers as gateways, and selecting one or more gateways from a list of desired partners. When two gateways consent to enabling a tunnel between the two gateways, the control system 175 may place each gateway on the partner list of the other gateway and provide the partner list to each gateway. Accordingly, the partner list may reflect the mutual desire of each gateway to enable a tunnel.

Moreover, the control system 175 may perform at least one or more of the following tasks, which are otherwise typically administered by the users enabling tunnels between gateways; coordinating one or more partner lists; administering the configuration of one or more virtual private networks established based on the enabled tunnels; monitoring the virtual private networks; controlling the virtual private networks; distributing to gateways information about changes in the configuration of the virtual private networks and/or other gateways; disseminating software for configuring gateways; providing an indication of a compromised private key; negotiating an encryption algorithm with gateways; negotiating an authentication technique with gateways; and recovering from a failure in the virtual private networks.

As previously discussed with reference to Figure 3, after a user desiring virtual private network services registers for secure services, the control system may assemble a disk image and provide the disk image to the

user for loading onto a computer and configuring the computer as a gateway. The gateway may then participate in a virtual private network established over a base network, such as the Internet.

Figure 5 illustrates an exemplary flow chart of the steps for establishing a virtual private network between the gateways identified by the user. Each of these steps will be discussed in further detail following the broad description of Figure 5.

Referring to FIGS. 4 and 5, the first gateway 450 may start with the disk image installed (step 510). The first gateway 450 may establish a connection to the control system 175 (step 520) and proceed to establish a first tunnel 425 to the control system 175 (step 530) through a communication channel, such as the communication channel 120 of FIG. 1. The second gateway 451 may also perform the steps 510-530 to establish a second tunnel 426 to the control system 175. Once the first and second tunnels are established, the control system 175 may exchange information with each gateway to further configure the gateways.

To enable a third tunnel 423 between the first gateway 450 and the second gateway 451 (step 540), the control system 175 may determine whether the first gateway 450 and the second gateway 451 have consented to enabling the third tunnel 423. This consent may be mutual and independent of the decision of the other gateways (not shown). For example, the control system 175 may determine the consent based on a list that includes desired partners for each of the gateways 450, 451. If the first gateway 450 and the second gateway 451 each consent to enabling of the third tunnel 423, the control system 175 may then enable the third tunnel 423 (step 540).

For example, to enable the third tunnel (step 540), the control system 175 may perform one or more of the following: update the partner lists of the first gateway 450 and the second gateway 451 to reflect mutual consent; provide an indication that a tunnel between the first and second gateways 450, 451 is authorized; provide real IP addresses for each of the gateways to permit a connection through a base network, such as the Internet; provide the virtual IP address of each gateway to the other gateway to enable a tunnel

between the gateways; facilitate the establishment of one or more tunnels by providing out-of-band signaling to the first gateway 450 and the second gateway 451 through the first tunnel 425 and the second tunnel 426, respectively; determine one or more partner lists for one or more gateways 450, 451; provide configuration information for the network and/or for each gateway; exchange control information with the first gateway 450 and the second gateway 451 on the first tunnel 425 and the second tunnel 426, respectively; negotiate an encryption algorithm with each gateway; and negotiate an authentication technique. Moreover, the control system 175 may also monitor the status and performance of the tunnels established through the communication channel 120 (step 550).

FIG. 6A shows a third exemplary network 600 in accordance with an embodiment of the present invention. The network 600 may include one or more local area networks (LANs) 660, 661, a first, second, and third gateways 650-652, the Internet 620 and/or Intranet access (not shown), and a network operations center 610.

The LANs 660, 661 may be similar to the LANs 160, 161 of FIG.1. The Internet 620 and/or Intranet access may include features similar to the communication channel 120 of FIG. 1. Moreover, the gateways 650-652 may each include information and program code for implementing one or more virtual private networks over the Internet 620. Furthermore, the first and second gateways 650, 651 may interface the LAN 660, 661 and the network 600 whereas the third gateway 652 may be configured as a stand-alone node interfacing only the network 600.

In the embodiment of FIG. 6A, the network operations center 610 may determine a virtual address for each gateway desiring to participate in one or more virtual private networks established through a base network, such as the Internet 620. Consequently, each gateway may be provided two addresses—a real or public address and a virtual address. The virtual address, which may be in an IP format, may be used by the gateways to establish one or more tunnels with each other through a base network, such as the Internet 620 and may be routable only through the established tunnels. This virtualized

-25-

addressing may provide virtual connectivity through the Internet 620 and may allow routing of virtual addresses from one address to another. Moreover, this virtualized addressing may facilitate network address translation, port address translation, IP masquerade, and/or IP connection sharing during the process of routing as well as during the dynamic assignment of addresses. Although a virtual address may be used by a gateway to establish one or more tunnels to form a virtual network and/or virtual private network, the network operations center 610 may alternatively provide to each gateway any other address that is capable of enabling any other networks established through or over a base network, such as the Internet 620.

Based on the virtual addresses determined by the network operations center 610 and provided to the gateways 650, 651, 652, one or more virtual private networks may be established over the Internet 620. For example, each gateway 650, 651, 652 may include a virtual device adapter (not shown), which may be capable of emulating the functions of a network interface card (NIC). Using the virtual device adapter, each gateway may route or forward information, such as packets through tunnels established with other gateways.

FIG. 6B shows the network 600 of FIG 6A from the perspective of virtual addresses and real or public addresses that are used by gateways 650-652 to route information, such as packets through tunnels established through the Internet 620, in accordance with an embodiment of the present invention. The gateways 650-652 may be assigned real IP addresses 601, 602, 603 and virtual IP addresses 604, 605, 606, respectively. Each real IP address, which may be assigned by, for example, an Internet Service Provider (ISP), may be routable through a base network, such as the Internet 620. On other hand, each virtual address, which may be assigned and provided by the network operations center 610, may be only routable through the tunnels enabled by the network operations center 610 and established through the Internet 620.

The solid lines connecting the gateways 650-652 represent the real IP connectivity between the machines. The real IP addresses 601-603 used by gateways 650-652, respectively, may interface the Internet 620 or a local area

network, such as LAN's 660 and 661. The dashed lines represent virtual connectivity provided by the virtual IP addresses 604-606. Each gateway may include at least one virtual device adapter with a corresponding virtual IP address. For example, a virtual device adapter (not shown) may be included at each end of a tunnel 699 established between the first gateway 650 and the second gateway 651. Each virtual device adapter may have the corresponding virtual IP address for its gateway. For example, the virtual device adapter for the first gateway 650 may have a virtual IP address of 10.0.1.1 (shown as 604), and the virtual device adapter for the second gateway 651 may have a virtual IP address of 10.0.1.2 (shown as 605).

In one embodiment, the network operations center 610 may provide to each gateway a virtual IP address during the initial configuration of the gateway. The network operations center 610 may then store the virtual IP address of the gateway with the gateway's name and the authentication information, such as a shared secret for that gateway. To enable a tunnel between two gateways that mutually consent to the tunnel, the network operations center 610 may provide each gateway the virtual IP address of the other gateway.

Packets addressed with a virtual IP address may be transported between the gateways through tunnels established through a base network, such as the Internet 620. For example, when a pair of gateways (e.g., 650 and 651) consents to enabling a tunnel (e.g. tunnel 699) between the gateways, the network operations center 610 may provide the virtual addresses for each gateway to the other gateway to enable the tunnel between the gateways.

Before the first gateway 650 sends a packet with an encrypted payload through a tunnel to the second gateway 651, the virtual device adapter may add the virtual addresses of the second gateway 651 and the first gateway 650 to the packet. For example, the virtual device adapter may add a source virtual address of 10.0.1.1 (shown as 604) and a destination virtual address of 10.0.1.2 (shown as 605) to a packet from the first gateway 650 to the second gateway 651. The first gateway 650 may then take the virtualized packet and

-27-

encapsulate the virtualized packet within another TCP/IP packet with real source and destination addresses, such as a source address of 193.168.100.5 (shown as 601) for first gateway 650 and a destination address of 193.11.10.3 (shown as 602) for second gateway 651. The encapsulated packet may then be routed based on the real destination address of 193.11.10.3 through the Internet 620 until the packet reaches the real destination address.

When the encapsulated packet arrives at the destination address, the second gateway 651 may remove the real TCP/IP addresses, leaving a payload that includes an IP packet with the virtual source and destination addresses. The virtual device adapter within the second gateway 651 may recognize the virtual IP addresses, receive the packet with the virtual IP addresses (i.e., source and destination virtual addresses), and forward the packet to the second gateway 651 for additional processing, such as authenticating and/or decoding the encrypted payload of the packet.

In one embodiment, network operations center 610 may enable and administer one or more virtual private networks, such as tunnels established through the Internet 620. The network operations center 610 may include one or more processors that are distributed or co-located within substantially the same geographic area. For example, the network operations center 610 may be distributed along a communication channel (see, e.g., the communication channel 120 at FIG. 1), the Internet, and/or an Intranet.

The network operations center 610 may perform at least one or more of the following features: providing information and code for configuring processors, such as computers as gateways capable of participating in one or more virtual private networks established through the Internet 620; enabling the establishment of tunnels by providing an indication that a tunnel between two gateways is authorized; determining one or more partner lists for gateways; administering the configuration of the virtual private networks; detecting and resolving virtual and real IP address conflicts; monitoring the virtual private networks; controlling the virtual private networks; negotiating an encryption algorithm with each of the gateways; providing a virtual IP address

to each gateway; negotiating an authentication technique with each of the gateways; distributing changes to the configuration of the virtual private network; disseminating software updates to the gateways; providing an indication of a security problem (e.g., a compromised private key); and recovering the virtual private networks from failures.

Accordingly, a user's role is simplified to registering with the network operations center 610, providing configuration information about one or more of the desired gateways, loading program code onto one or more computers to configure them as gateways, and selecting one or more desired partners for establishing one or more virtual private networks over a base network, such as the Internet 620.

Referring back to FIG. 6A, the network operations center 610 may include a public web server 611, a tunnel interface module 612, a proxy module 613, a controller module 614, an administrative server 615, a database server 616, one or more firewalls 617, one or more switches 680, and a communication channel 681.

The public web server 611 may not authenticate the identity of those connected to the public web server 611, and thus, may not provide any measure of trust. Moreover, the public web server 611 may not provide encryption or privacy. But the public web server 611 may provide a user with a means of accessing the network operations center 610 to perform limited functions, including registering to enable and establish a virtual private network through the Internet 620.

For example, a user may register through the public web server 611 in a nonsecure manner. During initial registration, the network operations center 610 and/or the public web server 611 may present to the user a series of questions and receive responses to the question based on which the network operations center 610 may generate program code and information for configuring a computer as a gateway capable of participating in one or more virtual private networks established over the Internet 620. For example, this program code and information may be provided in the form of a disk image, which may be downloaded and installed in one or more computers to

configure them as gateways 650-652. Moreover, the public web server 611 may also include one or more of the following: marketing information, trouble ticket information, and other user information that may not require privacy and/or authentication. The public web server 611 may include a firewall 617 and other security devices to limit access to the switch 680 and the communication channel 681 in network operation center 610. In one embodiment, the Linux Ipchains utility may be used to manage the firewall 617.

The tunnel interface module 612 may include program code for establishing tunnels between the network operations center 610 and one or more of the gateways 650-652. The tunnel interface module 612 may also include a public addressable or routable IP address that permits establishing tunnels between the network operations center 610 and the gateways 650-652 through the Internet 620. Moreover, the tunnel interface module 612 may include a transmission control protocol (TCP) tunnel driver used to establish a TCP tunnel between the network operations center 610 and the gateways 650-652. For example, the tunnel interface module 612 may use the TCP tunnel driver to encapsulate packets for an IPSec tunnel within TCP packets. Although the TCP tunnel driver may encapsulate the IPSec tunnel, other encryption and/or tunnel software (e.g., a User Datagram Protocol (UDP) tunnel driver) may be used instead.

In one embodiment, the only processes that may be executed from the nonsecure side of the tunnel interface module 612 (i.e., the Internet side 620) may be those processes related to the TCP tunnel driver.

To enhance security, the tunnel interface module 612 may communicate with the other subsystems of the network operations center 610 in a limited manner. For example, the tunnel interface module 612 may provide a single control and monitoring port for exchanging messages with the controller module 614 and for exchanging secured sockets layer (SSL) messages with the administrative server 615. Further, the tunnel interface module 612 may use a firewall 617 and/or other security devices to limit access to the switch 680 and communication channel 681. The two-tier

-30-

structure with the tunnel interface module 612 connected through security devices, such as firewalls to the controller module 614 may provide enhanced security at the network operations center 610.

The proxy module 613 may include one or more processors, which may serve as a proxy for enabling one or more tunnels between at least two of the gateways 650-652, when the gateways are each not accessible behind a firewall, hiding their respective real IP addresses. Alternatively, the proxy module 620 may be located within one of the gateways 650-652 or at a third party website hosting the proxy module 613.

The controller module 614 may include one or more processors, which may receive the control information provided by each of the gateways 650-652. The control information provided by each of the gateways 650-652 may also include monitoring information. The controller module 614 may also authenticate the identity of a gateway, determine that tunnels are authorized according to each gateway's list of desired partners, and add partners to each gateway's partner list.

The administrative server 615 gathers information and then may store gathered information in the database server 616 including, for example, a tunnel database that includes a list of tunnels that are active on the network 600; a predefined rule or trigger that indicates when a new tunnel request is made for a tunnel that already exists and is active in the tunnel database; a database with authentication information capable of authenticating the identity of each of the gateways 650-652 participating in the network 600. For example, the database server 616 may store for each gateway the authentication information in the form of a shared secret (e.g., a bit string and/or a public key) that authenticates the identity of a gateway seeking to establish a tunnel to the network operations center or another gateway. When the shared secret stored in the database server 616 matches the shared secret presented by the gateway to the network operations center 610, the gateway may be authenticated.

While encryption techniques may make communications private, authentication techniques may allow communicating parties to verify each

other's identity and the authenticity of the exchanged information.

Authentication serves to provide a level of trust so that users in a virtual private network may be confident about the authenticity of the exchanged information. Authentication may be established using a variety of security techniques including, for example, a signature, a digital signature, a digital certificate, a hash code, a password, and/or any other approach that may be used to establish identity of a user or computer.

The database server 616 may perform one or more of the following: storing customer information; storing the disk image described above; generating reports, such as alarm reports, activity reports, and/or other reports for administering virtual private networks established through the Internet 620; and storing monitoring information associated with the virtual private networks.

The firewalls 617 may include one or more processors which may selectively limit the type of information reaching communication channel 681 and switch 680. For example, the firewalls 617 may only permit entry of TCP commands to a specific port number. Moreover, the firewalls 617 may be implemented as a stand-alone device, software, firmware, and/or implemented as part of another processor, router, gateway, and/or any other device capable of performing the functions of a firewall.

The switches 680 switch information or traffic (e.g., datagrams, packets, or cells) between one or more of the subsystems 611-616 of the network operations center 610. The switches 680 may be implemented with one or more processors, a router, a switch, and/or any other communication device capable of switching and/or routing information to the appropriate subsystem within the network operations center 610.

The subsystems 611-616 of the network operations center 610 may be distributed along the communication channel 681 that connects the subsystems. The communication channel 681 may include one or more of the features and functions described above with respect to the communication channel 120 of FIG. 1.

-32-

FIG. 7 shows a flowchart of the steps performed for registering a gateway. A user, such as an administrator may register a gateway with the network operations center 610. A computer may connect through a gateway 650 to the Internet 620 and the public web server 611 of the network operations center 610 (step 710). Alternatively, a computer may connect directly to the Internet 620 and the public web server 611. The user of the computer, who may function as an administrator of the gateway 650, may provide registration information (step 720) to the public web server 611. The public web server 611 may then store the registration information (step 730) in, for example, the database server 616. The initial registration information may include preliminary configuration information, such as the number of gateways, billing information, and the administrator's name and email address.

Since the initial connection between the user's computer and the network operations center 610 may be a nonsecure connection, it may be desirable to limit the initial registration information to a minimum (e.g., the registration information provided above in step 720) to enhance security. This initial registration information may include the minimum amount necessary to create program code and information needed to configure a processor such that the configured processor is capable of contacting the network operations center 610 over a secure connection (e.g., a tunnel) established over the Internet 620 to obtain additional configuration information. Accordingly, once the user is able to communicate with the network operations center 610 through the secure connection, the user may then provide additional registration information. This additional information may be needed to complete the process of configuring the processor as a gateway. Further, this additional information may include, for example, the number and names for the gateways.

Once the processor is configured as a gateway, the network operations center 610 may prevent the gateway from connecting to the public web server 611 when exchanging additional information with the network operations center 610. For example, after a configured gateway contacts the network

-33-

operations center 610, the network operations center 610 may reroute any connections to the public web server 611 to the tunneling interface 612, where a secure tunnel is established for exchanging additional configuration information and code to complete the configuration of the gateway.

For example, during the user's first session with the public web server 611 of the network operations center 610, the user may connect to the network operations center using a browser configured with the Secure Sockets Layer protocol (SSL). During this initial contact with the public web server, the network operation center 610 may limit the user's range of permissible functions to basic functions until a secure tunnel is established. In one embodiment, the user may be denied the privilege to change firewall rules, administer partner lists, show tunnel status, show partner list information, delete administrators, and/or define groups of gateways. These denied functions may only be performed through a secure and/or authenticated tunnel to the network operation center 610.

FIG. 8 is an exemplary flow chart depicting the steps for configuring a gateway. The user may provide administration information (step 810); create an administrator login (step 820); create a password for the administrator's login (step 830); provide information describing at least one of the gateways 650-652, LAN 660, 661, Internet 620, and/or other information necessary to configure a gateway capable of participating in one or more virtual private networks established over the Internet 620 (step 840); and provide a name for each of the gateways 650-652 (step 850). The administrator may be a user with the authority to establish one or more virtual private networks over the Internet 620. The steps of FIG. 8 may be performed in a secure manner when the user uses one or more of gateways 650-652 to connect to the network operations center 610 and to establish a tunnel with the network operations center 610.

To provide administrator information (step 810), the user may use gateway 652 to connect to the network operations center 610 through the Internet 620. The user may provide the public web server 611 of the network operations center 610 with sufficient information for registering an

-34-

administrator including, for example, the administrator's name, log-in, password, e-mail address, pager, and phone number. In the exemplary embodiment of FIG. 6A, the public web server 611 may collect and store this information in database server 616. After the user provides this information (step 810), the network operations center 610 may create an administrator login (step 820), providing the user with the capability to configure and administer one or more virtual private networks over the Internet 620.

To create passwords (step 830), the user may select a login name and password for administration of the virtual network, such as a virtual private network for the gateways 650-652. The user may create a login and password for more than one administrator of the virtual private network to permit other users to login, create, administer, and download a disk image for configuring the virtual private network including the gateways. Furthermore, another user name and password may be created for access to a customer support function at the network operations center 610.

In providing information about the gateways 650-652, LAN 661, 660, and/or other information for configuring and administering virtual private networks (step 840), the user may provide one or more of the following information: the IP address; subnet mask; domain name server address; and gateway IP address for each desired gateway. If a fixed IP address gateway is not used for each gateway 650-652, the administrator may indicate that a dynamic host control protocol (DHCP) is used. Moreover, the administrator may provide other information including, for example, the media access control (MAC) address for a gateway or a proxy server IP address. For example, the network operations center 610 may perform an auto-discovery process to determine certain information about the administrator's existing network configuration. For example, the network operator center 610 may determine the IP address of a gateway by reading the source and destination address on a packet and determine whether the gateway is accessible behind a firewall by sending test packets to the gateway to see if the packets are rejected by the firewall.

To name each of the gateways 650-652 (step 850), the user may select a unique name for each of the gateways 650-652. Moreover, the user may select a name, such as a domain name for each of the configured virtual private networks. Furthermore, the user may select to use a two level naming hierarchy for each of the gateways 650-652. For example, a two level naming hierarchy may include, for example, domain_name.gateway_name or customer_name.organization_name.

Based on the information provided by the user, the network operations center may create and/or assemble program code and information for configuring a processor, such as a computer as a gateway capable of participating in one or more virtual private networks established over the Internet 620. For example, the network operations center 610 and, in particular, administrative server 615 may generate a disk image that includes the program code and information. The user may select to download the disk image during the initial session(s) with the network operations center 610. Alternatively, the user may select to download the disk image at a later session. The user may also select to receive the disk image in the form of a diskette; may select to store the disk image at the network operations center 610; and may permit one or more gateways 650-652 to download the disk image after the user's initial session with the network operations center 610.

FIG. 9 is an exemplary flow chart of the steps performed by network operations center 610 to create code and information (see, also, FIG. 3 at step 330) for configuring a gateway. The administrative server 615 in the network operations center 610 may gather the information previously provided by the user (step 910); create a disk image file (step 920); encrypt the disk image file (step 930); and send the disk image to the user (step 940).

To gather the information provided by the user (step 910), the administrative server may retrieve the information previously provided by the user (see, e.g., FIGS. 7 and 8) and store the information in the database server 616 of the network operations center 610. The administrative server 615 may then use this information to create a program code for configuring a

THIS PAGE BLANK (USPTO)

initiate IPSec tunnels through the TCP tunnels to the tunnel interface module 612 (step 1060). The tunnel interface module 612 may authenticate the identity of the gateway (step 1070); update the tunnel database (step 1080); and establish a connection from the gateway to the controller module 614 (step 1090). The controller module 614 may then activate a control path (step 1096), which the network operations center 610 may use to exchange control information with the gateway.

As each gateway is configured, it may perform the steps of FIG. 10 to establish a tunnel with the network operations center 610 and exchange through the tunnel, control information, monitoring information, and additional configuration information, such as the latest partner list.

In step 1010, the user of the first gateway 650 may install the disk image, enabling the first gateway 650 to reboot and execute the program code resident on the disk image.

In step 1020, the user may reboot the first gateway 650 with the program code. One of ordinary skill in the art would recognize that the reboot may take various forms and may include a total reboot of the gateway or, alternatively, a warm reboot where the gateway loads the disk image without affecting the operation of the gateway. Moreover, one of ordinary skill in the art would also recognize that the disk image may also be loaded on a communication device (e.g., a router, a firewall, a wireless device, and etc.) and/or any other processor. Moreover, the rebooting step 1020 may also include running other software including, for example, an operating system, drivers, program code for IPSec tunnels, and/or software capable of providing the functions of a firewall. RFC-2401, R. Atkinson, The Internet Society (1998), titled "Security Architecture for IP," describes, *inter alia*, IPSec and is incorporated herein by reference in its entirety.

In step 1030, the first gateway 650 may configure its IP addresses for the appropriate subnet mask, domain name server, Internet/Intranet access device, and/or Dynamic Host Configuration Protocol (DHCP) server. Moreover, the first gateway 650 may initialize its internal routing table to a default state.

-38-

The first gateway 650 may start the gateway daemon (step 1040), which may execute some or all of the program code on the disk image. The gateway daemon may contact the network operations center 610 (including the tunnel interface module 612 step 1050) using a domain name server or an IP address to resolve the address of the network operations center 610.

After initial contact with the network operations center 610 is made, the gateway daemon may open a TCP connection to the tunnel interface module 612. With a TCP tunnel established, the network operations center 610 may provide the gateway daemon with an IP address, permitting the first gateway 650 to make an internal routing table entry. This routing table entry may permit the first gateway 650 to route, for example, traffic associated with controlling a gateway through the TCP tunnel to the network operations center 610 and tunnel interface module 612. The first gateway 650 may then communicate directly with the tunnel interface module 612 through the TCP tunnel.

In step 1070, the first gateway 650 and the gateway daemon running on the first gateway 650 may begin the process of authentication with the network operations center 610. For example, an Internet Key Exchange (IKE) may be initiated between the network operations center 610 and the first gateway 650. This is described in RFC-2409, D. Harkins et al., The Internet Society (1998), titled "Internet Key Exchange," which is incorporated herein by reference in its entirety. A key exchange, such as IKE may be implemented using the Free S/WAN program code available at the Free S/WAN website. Alternatively, a shared secret may be presented for authentication.

During authentication, the first gateway 650 presents a shared secret to the network operations center 610. The authentication may include presenting a shared secret to the network operations center. In one embodiment, a gateway presented a virtual IP address that included a shared secret. Alternatively, a public key exchange, such as the one provided by the IKE protocol may also be used to authenticate the first gateway 650 with the network operations center 610 and the tunnel interface module 612. Furthermore, the shared secret or public key may also be used when a

gateway authenticates with another gateway during the establishment of a tunnel between the two gateways.

Moreover, during the authentication process, the tunnel interface module 612 may verify the authenticity of the first gateway 650 with information previously stored (e.g., the shared secret or public key stored during registration) at the database server 616. For example, the gateway name, virtual IP address of the gateway, and shared secret may be stored in the database server 616 during the initial registration of the first gateway 650. When the stored shared secret matches the shared secret presented by the first gateway 650, the identity or authenticity of the first gateway 650 is established. Alternatively, other authentication techniques and/or public key exchange techniques may be used. Moreover, the authentication system may be eliminated in an environment where authenticity and trust are not a concern. Authentication using MD5 is described in RFC-1828, P. Metzger et al., (1995) titled "IP Authentication using Keyed MD5," which is incorporated herein by reference in its entirety. Accordingly, once the first gateway 650 is authenticated with the network operations center 610, the first gateway 650 may exchange information with the network operations center 610 in a secure manner through an IPSec tunnel. With the first gateway 650 authenticated, the network operations center 610 may update the tunnel database (step 1080) stored at database server 616.

The first gateway 650 may open a connection, such as a TCP connection to the controller module 614 (step 1090) using the gateway daemon. The TCP connection to the controller module may go through the TCP tunnel to the controller module 614. For example, the controller module 614 may permit a connection, such as a control path on a predetermined TCP port. The predetermined TCP port may be the only port accessible through the tunnel interface module 612. As a result, the gateway daemon may initiate the TCP connection through the TCP tunnel to the tunnel interface module 612, the switch 680, and one or more of the firewalls 617 to access the control path at the predetermined TCP port (e.g., port 500) of the controller module 614. This TCP connection between the controller module

614 and the gateway daemon may serve as the control path for exchanging control information.

Before establishing the TCP connection between the first gateway 650 and controller module 614, the network operations center 610 may perform a tunnel database lookup to ensure that the TCP tunnel is a pending tunnel and not an active tunnel. If the TCP tunnel is an active tunnel, the network operations center 610 may provide an alarm. If the TCP tunnel is listed as pending in the tunnel database, the network operations center 610 may establish the control path between the controller module 614 and the tunnel interface module 612.

The network operations center 610 may also implement alarms when predetermined events occur that suggest a possible security concern or risk. The network operations center 610 may generate an alarm when one or more of the following conditions exist: an unauthorized computer attempts to authenticate posing as an established gateway; a tunnel flood attack; a failure to authenticate a gateway; a loss of the control path to a gateway; an internal failure within the network operations center 610 or gateway; an IP address of a gateway changes (i.e., if DHCP is not being used); a MAC address of a gateway's network interface card changes; a spoofing attempt; an attempt to authenticate a non-existent or denied gateway; excessive traffic associated with control or monitoring information; a failed attempt to logon (e.g., multiple tries); performance overruns; and authorization failures.

When the control path is activated by the controller module 614 of the network operations center 610 (step 1096), the tunnel interface module 612 may exchange control information with the first gateway 650. Moreover, the network operation center 610 may communicate one or more of the following information with the first gateway 650 through the control path: the virtual IP address of each gateway on the partner list, the partner list, the network settings, media access control (MAC) addresses, IP addresses (e.g., the DHCP server address, the domain name server address, an Internet access device), a check sum, a shared secret, program code for providing, configuring, and/or controlling a firewall, DHCP server code, and a "cookie."

This communication may take place using XML files. An exemplary set of XML files is shown below in Tables 1-6.

In one embodiment, the network operations center periodically receives through the control path monitoring information from the first gateway 660, such as the number of active tunnels, up/down times for each tunnel, and ping time between tunnels (i.e., latency). The monitoring information may be exchanged using XML files.

When the control path is activated (step 1096), the first gateway 650 may notify each of the other gateways that are listed on its partner list. Although steps 1010-1096 are described above with reference to the first gateway 650, each of the one or more gateways 650-652 may also perform steps 1010-1096. For example, the first gateway 650 may notify the second gateway 651 that it seeks to establish a third tunnel. The first gateway 650 and the second gateway 651 may then proceed to establish the third tunnel, after the third tunnel is enabled by the network operations center 610. Alternatively, the network operations center may enable the third tunnel by authorizing the third tunnel before the first gateway 650 and the second gateway 651 establish the tunnel. Accordingly, the first gateway 650 and the second gateway 651 may exchange information in a private and trusted manner through the established third tunnel that is enabled by the network operations center 610. The details of establishing the third tunnel are provided below.

FIG. 11 illustrates two exemplary partner lists 1110 and 1120, in accordance with an embodiment of the present invention. Each gateway 650-652 may consent to enabling one or more tunnels with another gateway by providing the network operations center 610 with a list of desired gateways from which it consents to enabling one or more tunnels. The network operations center 610 may determine whether two gateways consent to enabling a tunnel between the two gateways. If so, the network operations center 610 may place each gateway on a partner list of the other gateway. Accordingly, the partner list may reflect the mutual consent of the two gateways to enable one or more tunnels between the two gateways. In the

-42-

embodiment of FIG. 11, the network operations center 610 may generate for the first gateway 650 a partner list that lists the second gateway 651 as a partner. Similarly, the network operations center 610 may generate for the second gateway 651 a partner list that also lists the first gateway 650. If this is the case, the first gateway 650 and the second gateway 651 may mutually consent to enabling one or more tunnels between the first gateway and the second gateway. As a result, the consent may be mutual in that each gateway consents to enabling one or more tunnels with other gateways. The consents may also be independent in that the first gateway 650 and the second gateway 651 may decide independently of each other.

The network operations center 610 may determine a partner list for each of the gateways enabled by the network operations center 610 and may store the partner list for each enabled gateway. For example, the network operations center 610 may store a partner list for each gateway in a database within the database server 616. This database may store each gateway's name with a corresponding partner list that includes each partner's virtual IP address, public portion of the public key, firewall information, and other stored information. As a result, the network operations center 610 may enable a tunnel between the first gateway 650 and the second gateway 651 by determining that each gateway consents to enabling the tunnel and providing sufficient information, such as a partner list that includes each partner's virtual IP address, public portion of the public key, firewall information, etc. to each gateway such that the gateways are capable of establishing the tunnel.

FIG. 12 shows an exemplary screen 1250 for adding a gateway to a virtual private network enabled by the network operations center 610. FIG. 12 shows that a user may use the screen 1250 to graphically select one or more gateways from which the user's gateway would accept one or more tunnels. The screen 1250 may be presented to the user during the initial configuration of the user's gateway or whenever the user seeks to add a gateway to the user's virtual private network. The network operations center 610 may determine whether a gateway is selected by the user also consents to enabling one or more tunnels to the user's gateway. If the network operations

-43-

center determines that the selected gateway and the user's gateway mutually consent, the network operations center 610 may place the selected gateway on a partner list for the user's gateway; place the user's gateway on the selected gateway's partner list, and add the selected gateway to the virtual private network depicted in FIG 12.

FIG. 13 is an exemplary flow chart depicting steps for establishing a tunnel between at least two gateways in the network 600 shown in FIG. 6A. A gateway may seek to establish a tunnel, such as an IPSec tunnel with another gateway that is behind a firewall and is not accessible because the firewall selectively restricts information flowing to the gateway.

For example, after the first gateway 650 and the second gateway 651 have registered and established control paths with the network operations center 610, the first gateway 650 may seek to establish a tunnel to the second gateway 651. The network operations center 610 may enable the tunnel by providing the first gateway 650 with an indication that the second gateway 651 also consents to the enabling the tunnel. The network operations center 610 may acknowledge the mutual consent of the gateways by, for example, placing each gateway on the partner list of the other gateway.

The network operations center 610 may enable the tunnel by communicating the mutual consent to the first gateway 650 and the second gateway 651. This consent may be communicated in the form of providing a partner list to each gateway that consents to enabling the tunnel. The partner list may also include configuration information for each gateway listed in the partner list. The configuration information may provide sufficient information for establishing the tunnel and may include, for example, the following for each gateway listed on the partner list: a gateway name, a virtual IP address, a real IP address, and a shared secret for authentication with the network operations center and with other gateways enabled by the network operations center 610.

With the partner list, the network operations center 610 may also provide configuration information that includes, for example, firewall information indicating whether a gateway listed on a partner list is accessible

or whether the gateway is not accessible behind a firewall. For example, when the first gateway 650 contacts the second gateway 651 (step 1310) and attempts to establish a tunnel to the second gateway 651 (step 1320), the first gateway 650 may be notified by the network operation center 610 that the second gateway 651 is behind (i.e., not accessible behind) a firewall. In this example, the network operations center 610 may also provide the first gateway 650 with an indication that the first gateway is behind a firewall.

If the first gateway 650 is not behind a firewall, the first gateway 650, as the originating gateway for tunnel request, may determine whether the destination gateway (i.e., the second gateway 651) is behind a firewall (step 1340). If the destination gateway (i.e., the second gateway 651) is not behind a firewall (step 1340), the first gateway 650 may establish the tunnel to the second gateway 651 (step 1350) and exchange information with the second gateway 651 through the tunnel (step 1360). In one embodiment, the gateway with a lower IP address waits for a gateway with a higher IP address to establish a tunnel. In this embodiment, the gateway with the higher IP address is referred to as the originating gateway.

If the destination gateway (e.g., the first gateway 650) is not accessible behind a firewall (not shown) (step 1340), the originating gateway may wait for the destination gateway (e.g., the second gateway 651) to establish the tunnel (step 1370). When the second gateway 651 (i.e., the destination gateway) establishes the tunnel, the first gateway 650 and the second gateway 651 may exchange information through the established tunnel (step 1380).

If both the originating gateway (e.g., the first gateway 650) and the destination gateway (e.g., the second gateway 651) are not accessible behind firewalls (not shown) (steps 1330 and 1390), a direct tunnel between the originating gateway and the destination gateway may not be possible because the firewall may hide the real or public IP addresses of the originating gateway and destination gateway, respectively. As a result, the network operations center 610 may enable at the proxy module 613 a proxy (also referred to herein as a "Hairpin") (step 1391) to enable a tunnel between the first gateway and the second gateway 651 through the proxy.

-45-

When the Hairpin is enabled, the originating gateway that is not accessible behind a firewall and the destination gateway that is not accessible behind a firewall may exchange information through the Hairpin, bypassing the firewall of the other gateway (step 1392). The proxy module 613 may function as a Hairpin that may be enabled by the network operations center 610.

In one embodiment, the proxy module 613 may forward packets from one TCP port to another TCP port without examining the contents of the packets (e.g., reading the payload or decrypting the payload). Although the proxy module 613 shown in FIG. 6A may reside in the network operations center 610, the proxy module 613 may reside within any other device in the base network including, for example, another gateway. For example, if two gateways 650, 651 need a Hairpin, the third gateway 652 may serve as a Hairpin.

If the originating gateway is accessible a firewall (not shown) (step 1330) and the destination gateway is not behind a firewall (step 1390), the originating gateway may open a tunnel to the destination gateway (step 1393) and proceed to exchange information with destination gateway (step 1395) through the established tunnel.

FIG. 14 depicts a tunnel 1430 established between a first gateway 1410 and a second gateway 1420, in accordance with the steps depicted in the flow chart shown in FIG. 13. To establish the tunnel 1430, the first gateway 1410 may contact the second gateway 1420 (step 1310) and attempt to establish the tunnel 1430 to the second gateway 1420 (step 1320). In the embodiment of FIG. 14, the second gateway 1420 appears on the partner list of the first gateway 1410 and the second gateway 1420 may include the first gateway 1410 on its partner list. In this embodiment, neither the first gateway 1410 (i.e., the originating gateway) nor the second gateway 1420 (i.e., the destination gateway) is behind a firewall (steps 1330 and 1340). The first gateway 1410 may then establish the tunnel to the second gateway 1420 (step 1350) and proceed to exchange information with the second gateway 1420 through the established tunnel 1430 (step 1360).

-46-

Although the second gateway 1420 is not shown as being behind a firewall in FIG. 14, the second gateway 1420 may alternatively be placed behind a firewall. If the second gateway 1420 is placed behind a firewall (step 1340) and the second gateway is not accessible behind the firewall, the originating gateway (i.e., the first gateway 1410) may wait for the destination gateway (i.e., the second gateway 1420) to establish the tunnel 1430 (step 1370). While the originating gateway waits for the destination to establish the tunnel, the second gateway 1420 establishes a tunnel to the first gateway 1410 since the first gateway 1410 is accessible because it is not behind a firewall.

FIG. 15 illustrates a network 1500 that includes a first gateway 1510, a second gateway 1530, a network operations center 610, a proxy module 1520, a first tunnel 1532, a second tunnel 1531, and a control module 614. The gateways 1510 and 1530 are each behind firewalls 1590, 1591, respectively, that selectively restricts access to each of the gateways 1510, 1530. In this embodiment, the proxy module 1520 may reside in the network operations center 610. The first gateway 1510 may be the originating gateway that is not accessible behind a firewall 1590 (step 1330). Because the destination gateway (i.e., the second gateway 1530) may not be accessible behind a firewall 1591 (step 1390), the first gateway 1510 may not establish a tunnel directly to the second gateway 1530 and instead may use the proxy module 1520 as a Hairpin, bypassing the firewall 1591 of the second gateway 1530.

To enable the Hairpin (step 1391), the first gateway 1510 may use the configuration data provided by the network operations center 610 to determine that the second gateway 1530 is not accessible behind the firewall 1591. Alternatively, the first gateway may determine that the second gateway 1530 is not accessible behind the firewall 1591 through other means, such as sending packets to a real IP for the second gateway 1530. The first gateway 1510 may contact the controller module 614 to request enabling a tunnel to the second gateway 1530. The controller module 614 may then send a

-47-

message to the proxy module 1520 to enable a Hairpin for the first gateway 1510 and the second gateway 1530.

The proxy module 1520 may allocate a TCP port at the proxy module 1520 for the first gateway 1510 and another TCP port for the second gateway 1530. The proxy module 1520 may then provide the first gateway 1510 with the TCP port information and provide the second gateway 1530 with the other TCP port information. The proxy module 1520 may then initiate a TCP forwarding process that listens to both TCP ports allocated to the first gateway 1510 and the second gateway 1530, respectively. The controller module 614 may then proceed to inform the first gateway 1510 through the control path to establish a tunnel 1531 to the proxy module 1520 at the IP address of the proxy module 1520 and at the TCP port previously allocated to the first gateway 1510. The controller module 614 may also inform the second gateway 1530 to establish a separate tunnel 1532 to the proxy module 1520 at the IP address and at the TCP port allocated to the second gateway 1530.

The first gateway 1510 may then proceed to open a TCP connection to the TCP port previously allocated to the first gateway 1510 at the proxy module 1520. Similarly, the second gateway 1530 may open a TCP connection to the TCP port previously allocated to second gateway 1530 at the proxy module 1520. The proxy module 1520 may use the TCP protocol to forward TCP packets received from the first gateway 1510 to the second gateway 1530 and forward TCP packets received from the second gateway 1530 to the first gateway 1510. In the embodiment of FIG. 15, a tunnel from each of the gateways 1510, 1530 to the network operations center 610 may provide out-of-band signaling to enable the Hairpin at the proxy module 1520.

Accordingly, the proxy module 1520 may provide the capability to establish a tunnel between the first gateway 1510 and the second gateway 1530 by bypassing their respective firewalls 1590, 1591. Since firewalls may be configured to allow TCP traffic to originate from behind a firewall (i.e. outbound) but not allow arbitrary TCP traffic in (i.e. inbound), the first gateway 1510 and the second gateway 1530 may both send their respective TCP traffic to the proxy module 1520. Using TCP forwarding, the proxy module

-48-

1520 may act as a proxy to enable the exchange of information through a Hairpin even when the originating gateway and the destination gateway are both behind firewalls that selectively restrict access to the originating and destination gateways.

The network operations center 610 may control a firewall that selectively allows in-bound and out-bound traffic (e.g., firewalls 1590, 1591) based on a set of rules. For example, the rules may be used to restrict all in-bound and all out-bound traffic through the tunnels 1531, 1532.

Furthermore, the network operations center 610 may turn-off the rules, thus allowing an in-bound and out-bound traffic through the firewall. Although the firewalls shown in FIG. 15 reside outside of their respective gateways 1510 and 1530, the firewalls 1590 and 1591 may alternatively reside in their respective gateways 1510 and 1530.

If the network operation center 610 allows in-bound and out-bound traffic through the firewalls 1591, 1592 based on a set of rules, the firewalls 1590, 1591 may each be "on" and may filter packets received from the client side of their respective gateways and the tunnel side of their respective gateways. In this mode, by default, outgoing TCP, UDP, and Internet Control Message Protocol (ICMP) traffic originating on the client side may be allowed to reach the tunnel side. Similarly, the associated return packets from the tunnel side may be allowed to reach the client side. Furthermore, ICMP ping, traceroute traffic, and Domain Name Server (DNS) response traffic (i.e., UDP traffic including responses to a DNS request that originates from a processor on the client side) may also be allowed to reach the client side from the tunnel side. Finally, all other traffic originating from any other source on the tunnel side may be blocked.

The network operations center 610 may prompt the user of the network 1500 to select particular protocols that pass from the tunnel side to the client side. For example, the network operations center 610 may prompt a user of the gateway 1510 to select additional protocols, such as file transfer protocol (FTP), hypertext transfer protocol (HTTP), secure socket layer protocol (SSL), mail retrieval protocols (e.g., POP3), simple mail transfer protocol (SMTP),

and remote login protocol (e.g., TELNET). The user may also be prompted to create additional firewall parameters, such as selecting an allowable protocol, port, and direction for packets allowed through a firewall. For example, when a user is prompted to select an allowable protocol, port number, and direction, the user may select a TCP port number at a gateway to serve as a destination port for all TCP/IP packets received from the tunnel side of the firewall.

In another embodiment, a firewall maybe "on" and all client side and tunnel side packets other than packets destined for a tunnel enabled by the network operations center 610 are blocked.

The network operations center 610 may also turn-off the rules associated with a firewall. In this mode, the firewall is essentially "off" and packets are allowed to reach the client side of the firewall from the tunnel side.

FIG. 16A shows a network 1600A that includes a gateway 1610, a tunnel 1620, and the network operations center 610. The network operations center 610 may include a tunnel interface module 1630, a controller module 640, a database server 616 with an administrative server 1618. The gateway 1610 may include a gateway daemon as described above. The gateway 1610 may include a TCP tunnel driver that generates TCP packets forming a TCP tunnel that encapsulates an IPSec tunnel; an IPSec program code, such as the IPSec program code provided by Free S/Wan to establish the IPSec tunnel; and a virtual device adapter that functions as a virtual network interface card for recognizing a virtual IP address corresponding to the gateway 1610. The tunnel 1620 may include a data path for voice, video, and/or data and a control path for control and monitoring information.

FIG. 16B illustrates a network 1600B that includes a gateway 1610, a client 1615, a tunnel 1620, the network operations center 610, and a local area network 1617. The client 1615, which may include a processor such as a personal computer or any other processing device, may connect to the gateway 1610 through the local area network 1617. The gateway 1610 may then route the client's 1615 packets through the tunnel 1620 to a destination, such as the network operations center 610. Alternatively, the gateway 1610

may route the client's 1615 packets to other gateways (not shown) through one or more tunnels that are enabled by the network operations center 610.

The client 1615 may also use a data path within the tunnel 1620 to retrieve administrative information from the administrative server 1618. Furthermore, a control path may also be established to the controller 640 through the tunnel interface module 1630. The control path may carry control information, such as out-of-band signaling information for enabling one or more tunnels from the gateway 1610. The control information may include, for example, a partner list exchanged between the network operations center 610 and the gateway 1610.

FIG. 17 is an exemplary flow chart for a protocol that may be implemented to communicate between the gateway 1610 and the network operation center 610 shown in FIG. 16A. The gateway 1610 may connect to the tunnel interface module 1630 in the network operations center (NOC) 610 using a TCP tunnel (step 1710) and provide to the tunnel interface mode 1630 a virtual IP address and shared secret to authenticate with the network operations center 610.

The tunnel interface module 1630 may use the virtual IP address of the gateway 1610 to search and retrieve a shared secret stored within the network operation center 610 (step 1720). The shared secret may consist of a simple password, a simple bit string, a public key, or an MD5 hash. Alternatively, a public portion of a Public-Private Key pair may be used for authentication. If the shared secret provided by the gateway 1610 is authentic and thus corresponds to the shared secret that is stored for the gateway 1610 (step 1730), the gateway 1610 may proceed to negotiate a TCP tunnel (step 1750) with the tunnel interface module 1630. If the shared secret is not authentic (step 1730), the tunnel interface module 1630 may disconnect the gateway 1610 (step 1740) and generate an alarm (step 1745).

To initialize the gateway (step 1760), the gateway 1610 may send to the tunnel interface module 1630 an initiation message that includes a public portion of the Public-Private Key (PPK) pair (i.e., generated with the RSA algorithm) and a name for the gateway 1610 (step 1750). In one

embodiment, program code compliant with RSA signature algorithm, such as RSAsig program code included in the Free S/WAN may be used to generate the public part of the key pair.

The network operations center 610 may determine whether to accept or reject a tunnel requested by the gateway 1610 by authenticating that gateway based on the shared secret.

The gateway 1610 may first request to sign-on to the network operations center 610 (step 1770). The network operations center 610 may then acknowledges the sign-on request. The gateway 1610 may then proceed to sign-on to the network operations center 610 (step 1770). This permits the gateway 1610 and the network operations center 610 to exchange configuration information (step 1780) including, for example, a partner list for the gateway 1610; virtual IP addresses and real IP addresses for the gateway 1610, network operations center 610, and any other gateways on the partner list for the gateway 1610; and/or public key information for authenticating the gateway 1610 with other gateways and the network operations center 610. In one embodiment, the configuration information is exchanged using XML files. Further, as the configuration of the gateway 1610 changes, the network operations center 610 may broadcast the configuration information to any other gateway listed on the partner list of the gateway 1610. Although FIG. 16A shows one gateway (e.g., the gateway 1610), a plurality of gateways (not shown) may connect to the network operations center 610 by performing the steps shown in FIG. 17.

Network operations center 610 may provide a means for a client 1615 to establish a connection via a tunnel of the gateway 1610 to the network operations center 610. Although FIG. 16B shows one client 1615, a plurality of clients (not shown) may be connected to the gateway 1610. If a plurality of clients are connected to the gateway 1610, each of the clients may access one or more tunnels to the network operations center 610 through the LAN 1617 and the gateway 1610. Accordingly, each of these clients may participate in the virtual private network of FIG. 16B.

-52-

Table 1 lists exemplary Extensible Markup Language (XML) name value pairs provided by the network operations center 610 for configuring a gateway. For example, a gateway may receive the configuration information for itself and for each gateway on its partner list. Moreover, a gateway may receive this XML information whenever the gateway is connected to the network operations center 610.

Referring to Table 1, the network operations center 610 may provide each gateway enabled by the network operations center with one or more of the following: a gateway name, a domain name for the virtual private network, a virtual Internet Protocol (IP) address, and a public IP address visible to the Internet 620. Moreover, the network operations center 610 may provide information describing one or more of the following: whether a gateway is accessible behind a firewall; a network configuration for a gateway; whether a dynamic host configuration protocol (DHCP) is used at a gateway; IP addresses of the primary and secondary domain name servers associated with a local area network interfacing a gateway; and an IP address of a local IP proxy device providing Internet access to a local area network interfaced to a gateway.

Table 2 lists exemplary XML name value pairs provided by the network operations center for configuring a media access layer interface (e.g., an Ethernet interface) at a gateway configured by the network operations center 610. Moreover, the gateway may receive this configuration information for itself and each gateway on its partner list. The network operations center 610 may provide a name for the media access interface, a local IP address for the media access interface, a gateway IP address for the media access layer interface associated with the gateway, a subnet mask for the media access layer interface associated with the gateway, and whether addresses for the media access layer interface are assigned using a DHCP.

Table 1: Configuration Information

<local computer information>	
computer_name	="org5"
domain_name	="bugwheat2"
virtualip_address	="10.0.11.130"
visibleip_address	="208.185.39.2"
firewall_in_place	="no"
network_config	="Inline (i.e., GATEWAY AND IAD)"
dns_from_dhcp	="no"
dns_primary	="10.10.10.2"
dns_secondary	="10.10.10.3"
Proxylp	="208.185.40.2"
</local computer information>	

Table 2: Local Interface Information

<local interface information>	
name	="eth0"
mac_layer_address	="00:90:27:EE:02:3B"
local_IP_address	="208.185.39.2"
gateway	="208.185.39.1"
subnet_mask	="255.255.255.0"
dhcp	="none"
</local interface information>	

Table 3 lists exemplary XML name value pairs provided by the network operations center for a local area network interfacing a gateway. Moreover, the gateway may receive the information for itself and each gateway on its partner list. For example, the network operations center 610 may provide a gateway with information describing a local area network, such as the local area networks 661, 660 interfacing each of the gateways 650, 651 shown in FIG. 6A. The XML name value pairs may include configuration information describing an IP address range for the local area network, describing one or more members of an Access Control List and whether to include a tunnel access privilege for each member of the Access Control List, and specifying a gateway address for a subnet interfacing the local area network.

Table 3: Local LAN Information

<local_LAN_Information><address range>	
startip_address_range	="208.185.49.1"
endip_address_range	="208.185.49.255"
Type	="included"
Gateway	=""
</address range>	
</local_LAN_Information>	

Table 4 lists exemplary XML name value pairs for cryptographic information provided by the network operations center 610 to a gateway. For example, a gateway may receive the cryptographic information for itself and each gateway on its partner list. The network operations center 610 may provide the cryptographic information to enable an encrypted information flow, such as an encrypted tunnel between the gateway and another gateway or the network operations center 610. This cryptographic information may include the type of encryption algorithm, format (e.g., standard associated with the algorithm), the key information for the algorithm (e.g., a public key), and other parameters for the encryption algorithm.

Table 4: Cryptographic Information

<cryptographic key>	
Kind	="PublicKey"
Type	="NOC's_Primary_Key"
Format	="RSA"
Encryption	="3DES"
Modulus	="0x ... 01"
modulus_bits	="1024"
public_exp	="0x03"
</cryptographic key>	

Table 5 lists exemplary XML name value pairs for firewall information provided by the network operations center 610 to a gateway. For example, the gateway may receive the firewall information for itself and each gateway on its partner list. The firewall information may modify and/or configure a firewall and may include rules for the firewall, such as the protocol type

permitted to traverse the firewall, a direction for the permitted protocol, allowable source and destination addresses (e.g., IP addresses and port addresses), a flag to enable the rules, a name for each rule, whether to accept packets from another firewall, and a number indicating the order in which rule is executed in a firewall.

In one embodiment, Tables 1-5 may be stored in the network operations center 610 and indexed according to gateway name and/or virtual IP address of a gateway.

Table 6 lists exemplary XML name value pairs for monitoring information received by the network operations center 610. In one embodiment, a gateway may provide monitoring information about tunnels enabled by the network operations center 610. This monitoring information may permit the network operations center 610 to monitor the latency and bandwidth associated with a tunnel. For example, every 5 minutes a gateway may send to the network operations center 610 information corresponding to the accumulated number of packets and bytes transmitted at the gateway; the accumulated number of packets received at the gateway; the minimum round-trip time, maximum round-trip time, and 5 minute average round-trip time (i.e., in milliseconds) for packets traveling between the gateway and each gateway on the partner list of the gateway.

Table 5: Firewall Information

<firewall rule>
protocol="tcp"
direction="in"
src_ip_mask="\$any"
src_port="1024:65535"
dst_ip_mask="\$1"
dst_port="21"
action="ACCEPT"
rule_number="1"
</firewall rule>

Table 6: Monitoring Information

<bandwidth>	
time_of_day	="1800Z"
interval	="5"
xmit_packets	="10000"
xmit_bytes	="160000"
rcv_packets	="5"
rcv_bytes	="40"
</bandwidth>	
<latency>	
tod	="1800Z"
interval	="500"
minimum	="50"
maximum	="500"
average	="100"
</latency>	

FIG. 18 shows a network 1800 including one or more client computers 1824, 1823 connected to a hub 1822 that interfaces a first gateway 1821.

The first gateway 1821 may interface the Internet 1840 through an Internet Access Device (IAD) 1820 (see, e.g., IAD1 in FIG. 18). The hub, gateway, and IAD may be in an in-line configuration. The network 1800 may also include one or more client computers 1834, 1833 that are connected to a hub 1832 interfacing a second gateway 1831. The second gateway 1831 may connect to a second IAD 1830 that provides access to the Internet 1840. The network operations center 610 may also interface the Internet 1840. Although the in-line configuration is shown, other configurations of the network 1800 may also be implemented. For example, the hub 1822 may connect directly to the IAD 1820 instead of connecting to the gateway 1821.

A tunnel may be enabled between the first gateway 1821 and the second gateway 1831 by the network operations center 610. Once established, the tunnel may pass through the IAD 1820, the Internet 1840, and an IAD 1830.

FIG. 19 is an exemplary flowchart for detecting address changes in the network 1800 shown in FIG. 18. The network operations center 610 may establish a first tunnel (not shown) to the first gateway 1821 and a second tunnel (not shown) to the second gateway 1831. Each of these tunnels may

be established through a base network, such as the Internet 1840 and may permit the network operations center 610 to exchange information including, for example, configuration information and/or monitoring information (see, e.g., Tables 1-6 above) with each of the gateways 1821, 1831 (step 1910).

To detect an address change (step 1920), the network operations center 610 may monitor the status of each gateway 1821, 1831 through the first and second tunnels, respectively. When a real or public address, such as a real or public IP address of gateway 1821 changes, the network operations center 610 may detect the change by determining that the first tunnel between the network operations center and the gateway 1821 is terminated. For example, when an Internet Service Provider (ISP) changes the public IP address associated with the IAD 1820, the network operations center 610 may drop the first tunnel to the first gateway 1821 and detect an address change at the first gateway 1821 (step 1920). The gateway 1821 may then use its new IP address (i.e., the new public IP address associated with the IAD 1820) to reestablish the first tunnel to the network operations center 610 (step 1930) by performing the steps shown in FIG. 17.

Before reestablishing the first tunnel, the network operations center 610 may first authenticate the gateway 1821 (e.g., using a public key for gateway 1821). Once the first tunnel is reestablished, the network operations center 610 may then store the new IP address associated with the gateway 1821 (step 1940) and inform other gateways as to the new IP address (step 1950).

When the public IP address (i.e., the real IP address) of the first gateway 1821 changes, the second gateway may 1831 also drop a third tunnel (not shown) between the second gateway 1831 and the first gateway 1821. The first gateway 1821 and the second gateway 1831 may then proceed to reestablish the third tunnel after the first gateway 1821 authenticates with the network operations center 610 and provides the public IP address to the network operations center 610. Although FIG. 18 is described in connection with only two gateways, additional gateways (e.g., the gateways 1810-1815)

may also be added to a virtual network, such as a virtual private network enabled by the network operations center 610.

In the embodiment of FIG. 18, when additional gateways (e.g., the gateways 1810 through 1815) are present and are included in the partner list of the first gateway 1821, the network operations center 610 may notify the additional gateways and/or computer 1862 as to the new public IP address of the first gateway 1821 (step 1950). For example, the network operations center 610 may broadcast the new public IP address to all of the gateways on the partner list of the first gateway 1821.

FIG. 20 is an exemplary flow chart for resolving IP address conflicts in a local area network interfacing a gateway. One or more client computers 1823, 1824 interfacing the first gateway 1821 may use IP addresses that are local or private and conflict with the local IP addresses of the client computers 1834, 1833 interfacing the second gateway 1831. For example, the locally assigned IP address associated with the clients 1823, 1824 of the first gateway 1821 may be identical and thus may conflict with the locally assigned IP addresses associated with the clients 1833, 1834 of the second gateway 1831. This address conflict may be possible because the IP addresses of the client computers 1824, 1823 may be private or local addresses that are routable within the local area network served by the first gateway 1821. Thus, if a client of the first gateway 1821 has the same IP address as a client of the second gateway 1831, information may not be routed between the clients with conflicting addresses. Although detecting such address conflicts may be applicable in various environments, when an extranet is established, a client may be external to an organization and thus may use a local address that is not compatible with the local addresses used on the organization's network, such as the organization's intranet, wide area network, or local area network.

An address conflict may be detected when the first gateway 1821 establishes a tunnel to the second gateway 1831 (step 2010). For example, the first gateway 1821 may receive an IP address range (see, e.g., Table 3) for the second gateway 1831 and determine that an address conflict exists. When an address conflict exists during the establishment of the tunnel

between the first gateway 1821 and the second gateway 1831, the first gateway 1821 may propose a first intermediate address space (step 2020). The second gateway 1831 may propose a second intermediate address space (step 2030). Each gateway 1821,1831 may then negotiate an intermediate address space that does not conflict with the range of local addresses for the clients interfacing the gateway.

To negotiate the first intermediate address space and the second intermediate address (step 2040), the second gateway 1831 may accept the first intermediate address space proposed by the first gateway 1821 if the second gateway 1831 finds the first intermediate address space acceptable. An address space may be acceptable when the proposed address space does not conflict with the second gateway's 1831 local addresses. If the second gateway 1831 does not find the first intermediate address space acceptable, the second gateway may request from the first gateway 1821 another first intermediate address space

If the first gateway 1821 finds the second intermediate address space proposed by the second gateway 1831 acceptable, the first gateway 1821 may accept the second intermediate address space. If the first gateway 1821 does not find the second intermediate address space acceptable, the first gateway 1821 may request another second intermediate address space from the second gateway 1831.

The first gateway 1821 and the second gateway 1831 may provide the range of addresses in the first intermediate address space and the second intermediate address, respectively, to the network operations center 610 (step 2050). For example, the first gateway 1821 and the second gateway 1831 may send the first and second virtual address intermediate address ranges to the network operations center 610 through the first and second tunnels, respectively.

To translate the address of a packet based on the first intermediate address space and the second intermediate address space (step 2060), the first gateway 1821 may convert addresses, such as the IP addresses of packets destined for the second gateway 1831 into the first intermediate

-60-

address space. The second gateway 1831 may then detect the packets addressed in the first intermediate address space. Similarly, the second gateway 1831 may convert the IP addresses of packets destined for the first gateway 1821 into the second intermediate address space. The first gateway 1821 may also detect the packets addressed in the second intermediate address space. Consequently, each gateway may be responsible for determining if a local address conflict exists with another gateway; resolving the address conflict; and translating addresses of the packets to and from the negotiated address space such that the translation is transparent to clients interfacing each gateway.

As additional gateways are added to the network 1800, each additional gateway may establish one or more tunnels enabled by the network operations center 610 (step 2010); propose and negotiate an intermediate address space(s) if an address conflict exists with another gateway (steps 2020-2040); send the intermediate address space(s) to the network operations center 610 (step 2050); and translate packets to and from the negotiated intermediate address spaces(s) (step 2060).

For example, when a third gateway 1810 is added to the network 1800, the third gateway 1810 may establish a tunnel enabled by the network operations center 610 to the first gateway 1821 (step 2010). The third gateway 1810 may also perform the steps 2020-2060 if an IP address conflict exists with the clients 1824, 1823 of the first gateway 1821. The third gateway 1810 may then establish a tunnel to the second gateway 1821 and perform steps 2020-2060 if an address conflict exists with the clients 1834, 1833 of the second gateway 1831. As each gateway is added to the network 1800, the added gateway may negotiate an intermediate address space with each existing gateway to resolve any local address conflicts. Accordingly, one or more intermediate address spaces may be negotiated in a pair-wise manner between pairs of gateways enabled by the network operations center 610.

FIG. 21 is a block diagram of another exemplary virtual private network 2000 enabled by the network operations center 610. The network 2000 may include a first computer 2100, a second computer 2200, a network operations

center 610, and a gateway 650 connected to a local area network 660 that includes one or more host or client computers 2662, 2663 and servers 2661, 2664. Moreover, the network 2000 may include one or more tunnels 2300, 2700, 2800 enabled by the network operations center for exchanging information between first computer 2100, second computer 2200, and gateway 650 and one or more tunnels 2400, 2500, and 2600 for exchanging information including configuration information and/or monitoring information (see, e.g., Tables 1-6) with the network operations center 610.

The host computers 2662, 2663 and servers 2661, 2664 may include computers similar to the host computers 154, 155. Furthermore, the servers 2661, 2664 may include servers that support printing, file sharing, electronic mail, image storage, video storage, application hosting, hosting network services, and other functions capable of being hosted on a server.

The first computer 2100 and the second computer 2200 may include processors, such as the host computers 154 and 155. In one embodiment, the first computer 2100 and the second computer 2200 may include a Windows™ operating system. Alternatively, the first computer 2100 and the second computer 2200 may include a Linux operating system. The first computer 2100 and the second computer 2200 may each be capable of establishing tunnels enabled by the network operations center 610.

The first computer 2100 and the second computer 2200 may be part of different subnets. If that is case, the network operations center 610 may assign a virtual IP address to the first computer 2100 and another virtual IP address to the second computer 2200 and resolve any local address conflicts using, for example, the steps shown in FIG. 20. Unlike the gateway 650 that routes information to host computers 662, 663 and servers 661, 664, the first computer 2100 and the second computer 2200 are stand-alone computers that may route packets to a tunnel 2300, 2700, 2800. Moreover, unlike the gateway 650 that may maintain a dedicated control path 2600 to the network operations center 610, the first computer 2100 and second computer 2200 may each connect to the network operations center 610 through tunnels 2400,

-62-

2500 when required to exchange control and/or monitoring information with the network operations center 610.

To enable a tunnel between the first and second computers 2100, 2200, the network operations center 610 may enable the tunnel 2300 between the first and second computers 2100, 2200 after the first and second computers 2100, 2200 perform the steps shown in FIG. 17 (see, e.g., steps 1710-1780). For example, in the embodiment of FIG. 21, the first computer 2100 may connect to the network operations 610 through the tunnel 2400 to exchange information, such as Tables 1-6 above. This information may include an indication that the first computer 2100 consents to the establishment of the tunnel 2300 with the second computer 2200. The second computer 2200 may also connect to the network operations 610 through the tunnel 2500 to exchange information and to indicate consent to enabling the tunnel 2300 between the first computer 2100 and the second computer 2200.

After indicating consent and the network operation center 610 enabling the tunnel 2300, the first computer 2100 and/or the second computer 2200 may disconnect the tunnels 2400, 2500 and establish the enabled tunnel 2300.

The first computer 2100 and/or the second computer 2200 may reconnect tunnels 2400, 2500 to the network operations center when necessary to exchange information. For example, if the address of the first computer 2100 changes, the second computer 2200 may drop the tunnel 2300 to the first computer 2100. The first computer 2100 may reestablish the tunnel 2400, authenticate with the network operations center 610, and provide a new IP address for the first computer 2100. Similarly, the second computer 2200 may reestablish the tunnel 2500, authenticate with the network operations center 610, and receive the new IP address for the first computer 2100. The first computer 2100 and second computer 2200 may then disconnect the tunnels 2400, 2500 to the network operations center 610 and reestablish the tunnel 2300.

-63-

If the first computer 2100 has limited communications capability, a user of the first computer 2100 may dial in to the network operations center 610 using a wired or wireless Internet connection to create the tunnel 2400. For example, the first computer 2100 may include a mobile processor, such as a laptop computer, a personal digital assistant, or an Internet appliance or any other processor capable of establishing one or more tunnel enabled by the network operations center 610. Using the first computer 2100, the user may exchange over the tunnel 2400 configuration information to enable one or more tunnels. The first computer 2100 may then disconnect the tunnel 2400 to the network operations center 610 and then establish a tunnel 2700 to the gateway 650 to exchange information securely with the host computers 2662, 2663 or servers 2661-2664 interfacing the gateway 650 through the local area network 660. As a result, the user of the first computer 2100 may exchange information securely in mobile and/or wireless environments.

In the embodiment of FIG. 21, the network operations center 610 may also enable one or more tunnels between networks that are administered independently of each other or are otherwise incompatible with each other, thus enabling instant extranets. For example, if a user seeks to provide limited access through gateway 650 to one or more resources of LAN 660, such as a server 2661, the gateway 650 may consent to enabling a tunnel from an external network or processor, such as computer 2100 and/or computer 2200. In one embodiment, the computers 2100, 2200 may not have addresses, protocols, or security features that are compatible with those of the gateway 650. Moreover, the gateway 650 may deny the computers 2100, 2200 access to other resources on the LAN 660, limiting access only to the server 2664 based on an access control list provided by the network operations center 610.

The above embodiments and other aspects and principles of the present invention may be implemented in various environments. Such environments and related applications may be specially constructed for performing the various processes and operations of the invention or they may include a general-purpose computer or computing platform selectively

-64-

activated or reconfigured by program code (also referred to as code) to provide the necessary functionality. The processes disclosed herein are not inherently related to any particular computer or other apparatus, and may be implemented by a suitable combination of hardware, software, and/or firmware. For example, various general-purpose machines may be used with programs written in accordance with teachings of the present invention, or it may be more convenient to construct a specialized apparatus or system to perform the required methods and techniques.

The present invention also relates to computer readable media that include program instruction or program code for performing various computer-implemented operations based on the methods and processes of the invention. The media and program instructions may be those specially designed and constructed for the purposes of the invention, or they may be of the kind well-known and available to those having skill in the computer software arts. Examples of program instructions include for example microcode, machine code, such as produced by a compiler, and files containing a high-level code that can be executed by the computer using an interpreter.

Other embodiments of the invention will be apparent to those skilled in the art from consideration of the specification and practice of the invention disclosed herein. It is intended that the specification and examples be considered as exemplary only, with a true scope and spirit of the invention being indicated by the following claims.

WHAT IS CLAIMED IS:

1. A method for enabling a network between a first processor and a second processor using at least one additional processor separate from the first processor and the second processor, the method comprising the steps of:
 - receiving, at the at least one additional processor, information indicating a consent on behalf of the first processor to enabling a tunnel between the first processor and the second processor;
 - receiving, at the at least one additional processor, information indicating a consent on behalf of the second processor to enabling a tunnel between the second processor and the first processor;
 - determining a first virtual address for the first processor and a second virtual address for the second processor such that the first and second virtual addresses uniquely identify the first and second processors, respectively, and are routable through the network; and
 - providing, by the at least one additional processor, to each of the first and second processors the first and second virtual addresses to enable one or more tunnels between the first and the second processors.
2. The method of claim 1, further comprising the step of:
 - establishing, by the first processor, one or more tunnels to the second processor using the first and second virtual addresses.
3. The method of claim 2, wherein said step of establishing further comprises the step of:
 - establishing each of the one or more tunnels as an encrypted tunnel.
4. The method of claim 3, wherein said step of establishing each of the one or more tunnels further comprises the step of:

-66-

establishing each of the one or more tunnels as an encrypted tunnel that is encapsulated within a protocol.

5. The method of claim 3, wherein said step of establishing each of the one or more tunnels further comprises the step of:

establishing each of the one or more tunnels as an encrypted tunnel based on an Internet Protocol Security (IPSec) tunnel.

6. The method of claim 1, wherein said step of receiving, at the at least one additional processor, information indicating a consent on behalf of the first processor, further comprises the step of:

receiving a name that identifies the second processor as consenting to enabling the one or more tunnels between the first and second processors.

7. The method of claim 1, wherein said step of receiving, at the at least one additional processor, information indicating a consent on behalf of the first processor, further comprises the step of:

receiving, at the at least one additional processor, information indicating the consent on behalf of the first processor to enabling one or more other tunnels between the first processor and other processors separate from the at least one additional processor and the second processor.

8. The method of claim 1, wherein said step of determining further comprises the step of:

selecting, at the at least one additional processor, each of the first and second virtual addresses from a predetermined address range.

9. The method of claim 8, wherein said step of selecting further comprises the step of:

defining each of the addresses in the predetermined address range as an address that is routable through the network when the network is enabled by the at least one additional processor.

10. A method for enabling a network between a first processor and a second processor using at least one additional processor separate from the first processor and the second processor, the method comprising the steps of:

establishing a first tunnel between the first processor and the at least one additional processor;

establishing a second tunnel between the second processor and the at least one additional processor;

determining, at the least one additional processor, whether the first and second processors mutually consent to enable a third tunnel between the first and second processors;

determining a first virtual address for the first processor and a second virtual address for the second processor such that the first and second virtual addresses uniquely identify the first and second processors, respectively, and are routable through the network; and

providing the first virtual address to the second processor through the second tunnel and the second virtual address to the first processor through the first tunnel after the at least one additional processor determines that the first and second processor mutually consent to enabling the third tunnel.

11. The method of claim 10, wherein said step of determining, at the at least one additional processor, further comprises the step of:

receiving, at the at least one additional processor, a consent from each of the first and second processors independently of each other.

12. The method of claim 10, further comprising the step of:
 - establishing, by the first processor, the third tunnel to the second processor using the provided first and second virtual addresses.
13. The method of claim 12, wherein said step of establishing the third tunnel, further comprises the step of:
 - establishing, by the first processor, the third tunnel to the second processor through a firewall associated with the second processor.

14. The method of claim 12, wherein said step of establishing the third tunnel, further comprises the step of:

establishing the third tunnel as an encrypted tunnel.

15. The method of claim 14, wherein said step of establishing the third tunnel as an encrypted tunnel further comprises:
 - establishing the third tunnel as an encrypted tunnel that is encapsulated within a protocol.

16. The method of claim 14, wherein said step of establishing the third tunnel as an encrypted tunnel further comprises the step of:

establishing the third tunnel as an encrypted tunnel based on an Internet Protocol Security (IPSec) based encrypted tunnel.

17. The method of claim 12, further comprising the step of:
 - excluding from the first and second tunnels information that flows from the second processor to the first processor through the established third tunnel.

-69-

18. The method of claim 12, further comprising the step of:
excluding from the first and second tunnels information that flows from
the first processor to the second processor through the established third
tunnel.

19. The method of claim 12, further comprising the step of:
interfacing the first processor to one or more other processors separate
from the first and second processors and the at least one additional processor
such that information is routed to the one or more other processors from the
second processor through the established third tunnel and the first processor.

20. The method of claim 12, further comprising the step of:
interfacing the first processor to one or more other processors separate
from the first and second processors and the at least one additional
processors such that information is routed to the second processor from the
one or more other processors through the first processor and the established
third tunnel.

21. The method of claim 12, further comprising the steps of:
providing, by the at least one additional processor, code and
information that uniquely identifies the first processor in the network; and
executing the provided code on the first processor to configure, based
on the provided information, the first processor as a gateway to one or more
other processors separate from the first and second processors and the at
least one additional processor; and

-70-

routing, at the configured first processor, information from the one or more other processors through the established third tunnel to the second processor.

22. The method of claim 21, wherein the step of providing further comprises the step of:

providing the code and the information on one or more computer readable media.

23. The method of claim 22, wherein the step of providing further comprises the step of:

downloading the code and the information from the at least one additional processor.

24. The method of claim 10, further comprising:
receiving, from the first processor at the at least one additional processor, information indicating a consent on behalf of the first processor to enable the third tunnel to the second processor.

25. The method of claim 24 wherein said step of receiving, from the first processor further comprises the step of:

receiving a name that identifies the second processor as consenting to enabling the third tunnel between the first and second processors.

26. The method of claim 25, wherein said step of receiving the name further comprises the step of:

receiving the name from the first processor through the first tunnel.

-71-

27. The method of claim 10, further comprising the step of:
receiving, at the at least one additional processor, information
indicating a consent on behalf of the first processor to enable one or more
other tunnels to other processors separate from the at least one additional
processor and the second processor.

28. The method of claim 10, wherein said step of determining a first
virtual address further comprises the step of:

selecting, at the at least one additional processor, each of the first and
second virtual addresses from a predetermined address range.

29. The method of claim 28, wherein said step of selecting further
comprises the step of:

defining each of the addresses in the predetermined address range as
an address that is routable through the network when the network is enabled
by the at least one additional processor.

30. The method of claim 10, further comprising the step of:
providing, from the at least one additional processor and through the
first tunnel, information to a firewall that selectively restricts a flow of
information into the first processor such that information flowing from the
second processor on the enabled third tunnel is allowed by the firewall into
the first processor.

31. The method of claim 10, further comprising the step of:
placing between the first processor and the second processor a firewall
to selectively restrict a flow of information into the first processor.

-72-

32. The method of claim 10, wherein said step of establishing the first tunnel further comprises the step of:

establishing the first tunnel through a proxy server placed between the first processor and the at least one additional processor.

33. The method of claim 10, further comprising the step of: monitoring, at the at least one additional processor, the first and second processors through the first and second tunnels.

34. A system for enabling a network between a first processor and a second processor, wherein the first and second processors are separate from said system, said system:

a tunneling interface that receives information indicating a consent on behalf of the first processor to enabling a tunnel between the first processor and the second processor, and receives information indicating a consent on behalf of the second processor to enabling a tunnel between the second processor and the first processor; and

a controller that determines a first virtual address for the first processor and a second virtual address for the second processor such that the first and second virtual addresses uniquely identify the first and second processors, respectively, and are routable through the network, and that provides to each of the first and second processors the first and second virtual addresses to enable one or more tunnels between the first and the second processors.

35. A computer program product for enabling a network between a first processor and a second processor using at least one additional processor

separate from the first processor and the second processor, the computer program product comprising code, said code comprising:

code, at the at least one additional processor, that receives information indicating a consent on behalf of the first processor to enabling a tunnel between the first processor and the second processor, and receives information indicating a consent on behalf of the second processor to enabling a tunnel between the second processor and the first processor;

code that determines a first virtual address for the first processor and a second virtual address for the second processor such that the first and second virtual addresses uniquely identify the first and second processors, respectively, and are routable through the network; and

code, at the at least one additional processor, that provides to each of the first and second processors the first and second virtual addresses to enable one or more tunnels between the first and the second processors.

36. A system for enabling a network between a first processor and a second processor, said system comprising:

at least one memory including

code that receives information indicating a consent on behalf of the first processor to enabling a tunnel between the first processor and the second processor and information indicating a consent on behalf of the second processor to enabling a tunnel between the second processor and the first processor,

code that determines a first virtual address for the first processor and a second virtual address for the second processor such that the first and

-74-

second virtual addresses uniquely identify the first and second processors, respectively, and are routable through the network, and

code that provides to each of the first and second processors the first and second virtual addresses to enable one or more tunnels between the first and the second processors; and

at least one processor, separate from the first and second processors, that executes said code.

37. A system for enabling a network between a first processor and a second processor, wherein the first and second processors are separate from said system, said system:

a tunneling interface that establishes a first tunnel between the first processor and the at least one additional processor and establishes a second tunnel between the second processor and the at least one additional processor;

a controller that determines whether the first and second processors mutually consent to enable a third tunnel between the first and second processors, determines that a first virtual address for the first processor and a second virtual address for the second processor such that the first and second virtual addresses uniquely identify the first and second processors, respectively, and are routable through the enabled network, and provides the first virtual address to the second processor through the second tunnel and the second virtual address to the first processor through the first tunnel after the controller determines that the first and second processor mutually consent to enabling the third tunnel.

38. A computer program product for enabling a network between a first processor and a second processor using at least one additional processor separate from the first processor and the second processor, the computer program product comprising code, said code comprising:

code that establishes a first tunnel between the first processor and the at least one additional processor and establishes a second tunnel between the second processor and the at least one additional processor;

code, at the least one additional processor, that determines whether the first and second processors mutually consent to enable a third tunnel between the first and second processors;

code that determines a first virtual address for the first processor and a second virtual address for the second processor such that the first and second virtual addresses uniquely identify the first and second processors, respectively, and are routable through the network; and

code for providing the first virtual address to the second processor through the second tunnel and the second virtual address to the first processor through the first tunnel after the at least one additional processor determines that the first and second processor mutually consent to enabling the third tunnel.

39. A system for enabling a network between a first processor and a second processor, said system comprising:

at least one memory including code comprising

-76-

code that establishes a first tunnel between the first processor and the at least one additional processor and establishes a second tunnel between the second processor and the at least one additional processor,

code, at the least one additional processor, that determines whether the first and second processors mutually consent to enable a third tunnel between the first and second processor,

code that determines a first virtual address for the first processor and a second virtual address for the second processor such that the first and second virtual addresses uniquely identify the first and second processors, respectively, and are routable through the network, and

code that provides the first virtual address to the second processor through the second tunnel and the second virtual address to the first processor through the first tunnel after the at least one additional processor determines that the first and second processor mutually consent to enabling the third tunnel; and

at least one processor, separate from the first and second processors, that executes said code.

40. A network comprising:

a first processor;

a second processor; and

at least one additional processor, separate from the first and second processors, wherein the at least one additional processor determines a first virtual address for the first processor and a second virtual address for the second processor such that one or more tunnels are enabled when the at

least one additional processor determines that the first and second processors mutually consent to enabling the one or more tunnels between the first processor and the second processor and provides the second virtual address to the first processor and the first virtual address to the second processor to enable the one or more tunnels.

41. The network of claim 40 further comprising:

a third processor, separate from the at least one additional processor and the second processor and placed between the first processor and the second processor such that the third processor selectively restricts into the first processor a flow of information on the enabled one or more tunnels.

42. The network of claim 40 further comprising:

one or more other processors, separate from the first and second processors and the at least one additional processors, that interface to the first processor such that information is routed to the second processor from the one or more other processors through the first processor.

43. A system for enabling a network between a first processor and a second processor using at least one additional processor separate from the first processor and the second processor, the system comprising the steps of:

means for receiving, at the at least one additional processor, information indicating a consent on behalf of the first processor to enabling a tunnel between the first processor and the second processor;

means for receiving, at the at least one additional processor, information indicating a consent on behalf of the second processor to enabling a tunnel between the second processor and the first processor;

means for determining a first virtual address for the first processor and a second virtual address for the second processor such that the first and second virtual addresses uniquely identify the first and second processors, respectively, and are routable through the network; and

means for providing, by the at least one additional processor, to each of the first and second processors the first and second virtual addresses to enable one or more tunnels between the first and the second processors.

44. A system for enabling a network between a first processor and a second processor using at least one additional processor separate from the first processor and the second processor, the system comprising the steps of:

means for establishing a first tunnel between the first processor and the at least one additional processor;

means for establishing a second tunnel between the second processor and the at least one additional processor;

means for determining, at the least one additional processor, whether the first and second processors mutually consent to enable a third tunnel between the first and second processors;

means for determining a first virtual address for the first processor and a second virtual address for the second processor such that the first and second virtual addresses uniquely identify the first and second processors, respectively, and are routable through the network; and

means for providing the first virtual address to the second processor through the second tunnel and the second virtual address to the first processor through the first tunnel after the at least one additional processor

determines that the first and second processor mutually consent to enabling the third tunnel.

45. A method for establishing a virtual private network on a base network, the method comprising:

providing code to be loaded onto a first computer at a first desired network site to facilitate the establishment of a first gateway on the first computer at the first desired network site;

providing code to be loaded onto a second computer at a second desired network site to facilitate a second gateway on the second computer at the second desired network site;

receiving, via the base network, and remote from the first and second network sites, information from the first and second computers;

establishing a virtual private network based on the received information; and

administering the virtual private network at a site remote from the first and second sites.

46. The method of claim 45, wherein the base network includes the Internet.

47. The method of claim 45, further comprising providing code to additional computers to facilitate the establishment of network sites, in addition to the first and second sites.

48. The method of claim 45, wherein code provided to facilitate the establishment of a gateway is transmitted to network sites via the Internet.

-80-

49. The method of claim 45, wherein code provided to facilitate the establishment of a gateway is transmitted to a network site by shipping the code on a physical storage medium.

50. The method of claim 45, further comprising adding an additional network site to the virtual private network after the virtual private network is established.

51. The method of claim 45, wherein a plurality of virtual private networks are administered over the Internet via the remote site.

52. A method for establishing a virtual private network (VPN) on a base network, said method comprising the steps of:

receiving, at a mediation point in the base network, a request for configuring a gateway; and

providing code to the gateway in response to the request such that the gateway is configured via the mediation point.

53. The method of claim 52, wherein the providing step comprises the substeps of:

downloading code from the mediation point;

booting the gateway from the downloaded code; and

establishing a tunnel from the gateway to the mediation point such that the gateway is configured via the mediation point.

54. The method of claim 52, further comprising the step of:

establishing one or more virtual private networks through the configured gateway.

55. The method of claim 54, further comprising the step of: administering the established virtual private network via the mediation point.

56. The method of claim 54, further comprising the steps of: monitoring the established virtual private networks via the mediation point.

57. The method of claim 56, wherein the monitoring step comprises the step of:

collecting, via the mediation point, state information associated with the established virtual private networks.

58. The method of claim 56, wherein the monitoring step comprises the substep of:

monitoring services provided by one or more network service providers.

59. The method of claim 56, wherein the monitoring step comprises the substep of:

monitoring performance of each of the established virtual private networks.

60. The method of claim 52, further comprising the step of monitoring the configured gateway.

61. The method of claim 52, further comprising the step of: administering the configured gateway via the mediation point.

62. The method of claim 52, further comprising the step of: adding features to the configured gateway via the mediation point.

-82-

63. The method of claim 52, further comprising the step of:
modifying a configuration of the configured gateway via the mediation point.
64. A method for managing a virtual private network on a base network, said method comprising the steps of:
providing via a mediation point in the base network, code for configuring one or more gateways in the base network;
establishing one or more private virtual networks through the configured gateways; and
managing the established virtual private networks via the mediation point.
65. The method of claim 64, wherein the providing step comprises the substep of:
interrogating one or more devices in the network to determine parameters for configuring the code.
66. A method for establishing a virtual private networks on a base network that includes gateways, said method comprising the steps of:
identifying, for at least one of the gateways, one or more other gateways that are authorized to establish one or more virtual private networks through the at least one gateway; and
establishing one or more virtual private networks through the at least one gateway only if the authorized gateways participate in the virtual private networks.

67. A method for establishing communication between two or more gateways that are located behind respective firewalls in a base network, said method comprising the steps of:

establishing a first tunnel from a first gateway to a mediation point in the base network;

establishing a second tunnel from a second gateway to the mediation point; and

routing, at the mediation point, information received through the first tunnel to the second tunnel and vice versa such that communication is established between the first and the second gateways.

68. A method for configuring a gateway in a base network, said method comprising the step of:

providing, via a mediation point in the base network, code for configuring one or more gateways in the base network such that the configured gateways are administered or monitored via the mediation point.

69. A method for managing one or more addresses in a base network that includes at least one virtual private network, said method comprising the steps of:

detecting, at a first gateway, a changed address of a device through which a virtual private network associated with the first gateway is established on the base network;

reestablishing, by the first gateway, a tunnel to a mediation point in the base network; and

storing, at the mediation point, the changed address of the device.

-84-

70. The method of claim 69, further comprising the step of:
notifying one or more other gateways that participate in the virtual
private network of the changed address of the device; and
reestablishing, from the other gateways, tunnels to the first gateway
using the address.

71. A method for managing a virtual private network on a base
network, said method comprising the steps of:
providing code for configuring one or more gateways through which
one or more virtual private networks is established in the base network; and
providing a mediation point in the base network for administering or
monitoring the virtual private networks.

72. A method for establishing a network, said method comprising
the steps of:

identifying, at a mediation point in the network, an identity of one
or more participants in the network;
notifying each participant via the mediation point of the identity
of other on-line participants;

establishing tunnels from each on-line participant to other on-
line participants based on information provided by the mediation point such
that each on-line participant has access to one or more resources of the other
participants.

73. A method for configuring a gateway in a network, said method
comprising the steps of:

providing code to be loaded onto a personal computer such that, after rebooting the personal computer, the computer establishes communication with a mediation point in the network to receive information for configuring the personal computer as a gateway.

74. A method for establishing communication between two or more gateways that are located behind respective firewalls in a network, said method comprising the steps of:

establishing a first tunnel from a first gateway to a site in the network remote from the first gateway;

establishing a second tunnel from a second gateway to the remote site; and

mediating, at the remote site, information received through the first and the second tunnels such that communication is established between the first and the second gateways.

75. A method for managing one or more addresses in a network comprising gateways, said method comprising the steps of:

identifying, in at least one gateway, one or more clients desiring to participate in the network; and

negotiating with one or more other gateways to allocate network addresses to the clients such that the clients are capable of participating in the network.

76. The method of claim 75, wherein the negotiating step comprises the step of allocating the network addresses via a mediation point in the network.

-86-

77. A method for managing one or more addresses in a base network that includes at least one virtual private network, said method comprising the steps of:

detecting, at a first gateway, a changed address of a device through which a virtual private network is established on the base network; and notifying, via a mediation point in the base network, one or more other gateways of the changed address such that the changed address is transparently administered by the mediation point.

78. A method for establishing one or more virtual private networks on a base network that includes gateways, said method comprising the steps of:

maintaining, in at least one gateway, a list of one or more gateways authorized to establish one or more tunnels through the at least one gateway; receiving, in the at least one gateway, a request from another gateway for establishing a tunnel through the at least one gateway; and consenting to the requested tunnel only if the another gateway is on the list, otherwise, denying the requested tunnel.

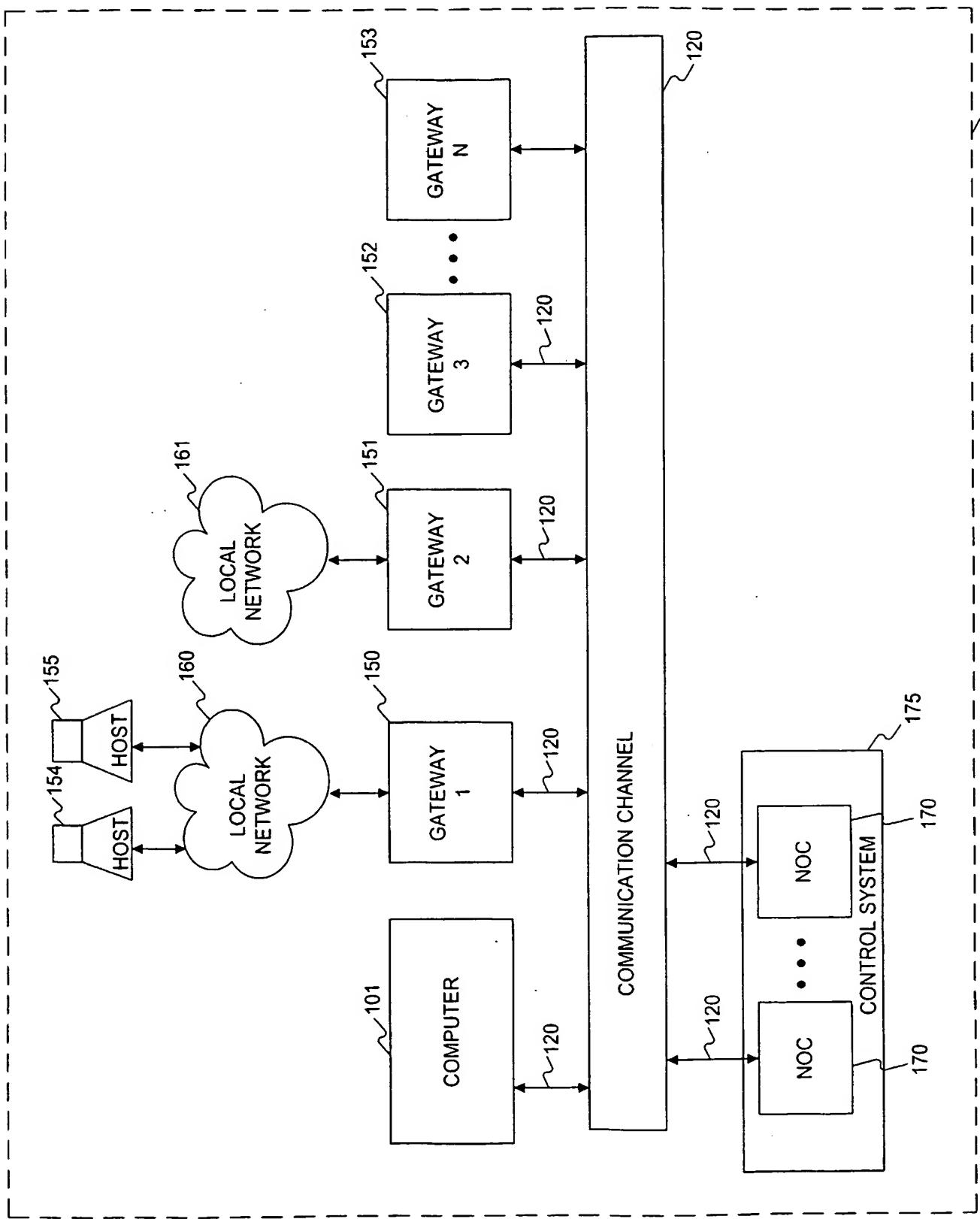


FIG. 1

THIS PAGE BLANK (USPTO)

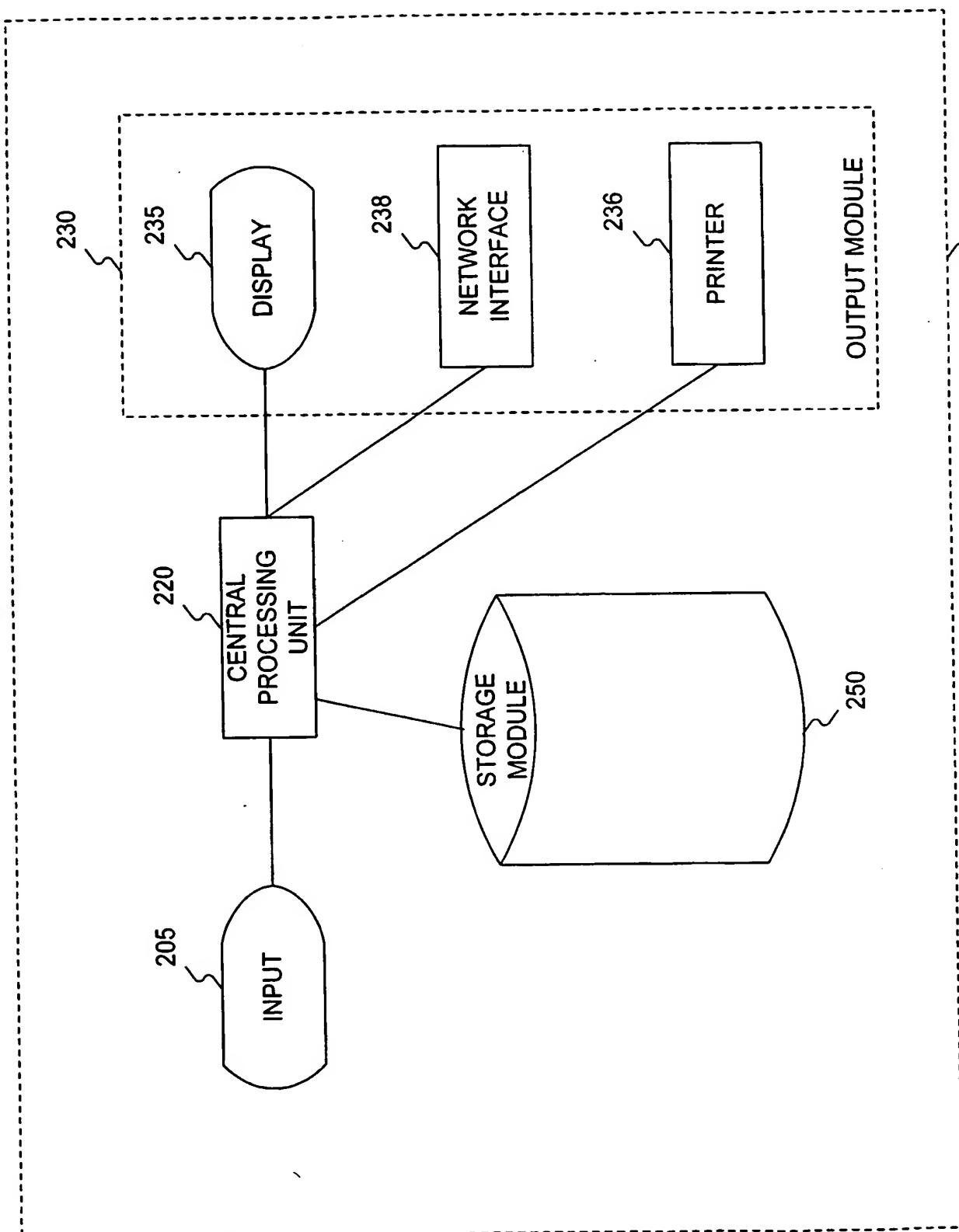
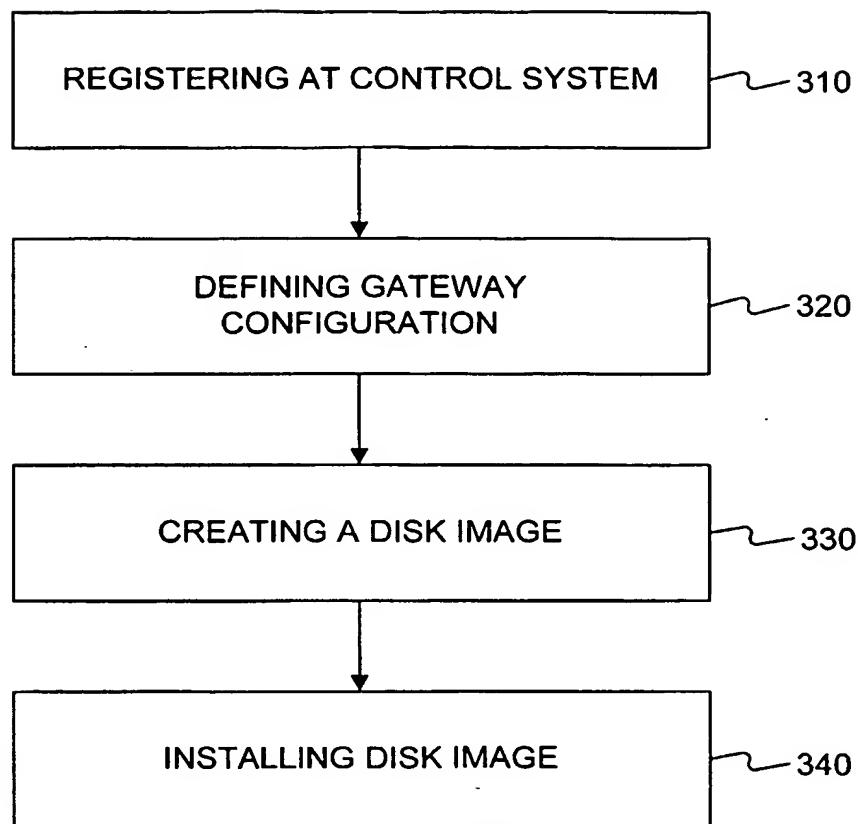
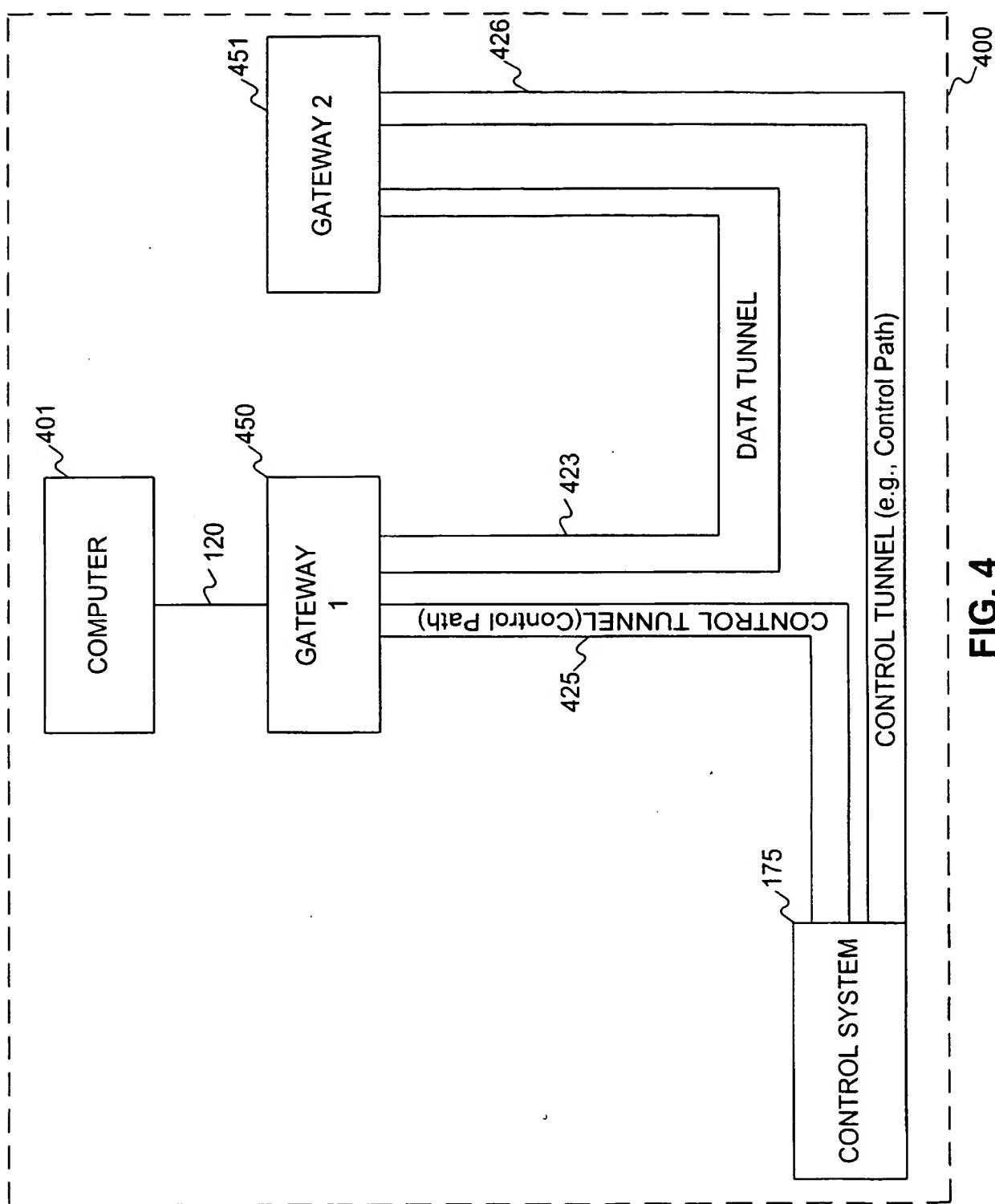


FIG. 2

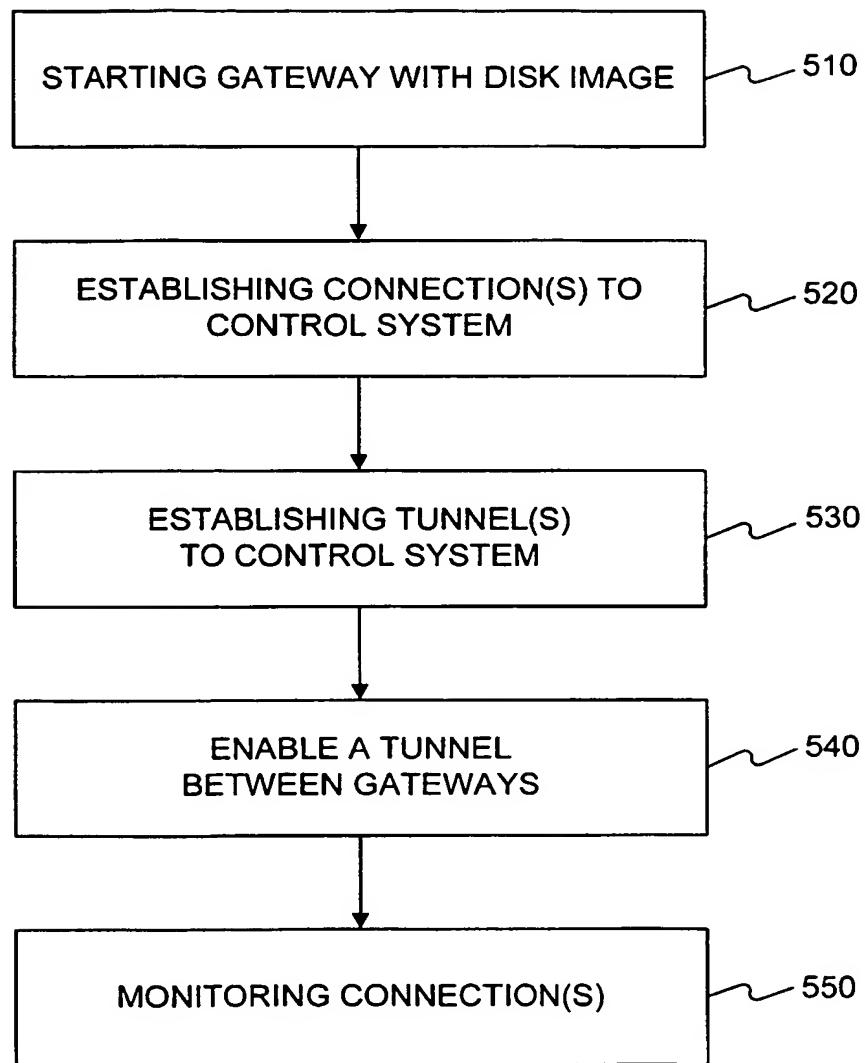
THIS PAGE BLANK (USPTO)

**FIG. 3**

THIS PAGE BLANK (USPTO)

**FIG. 4**

THIS PAGE BLANK (USPTO)

**FIG. 5**

THIS PAGE BLANK (USPTO)

6/23

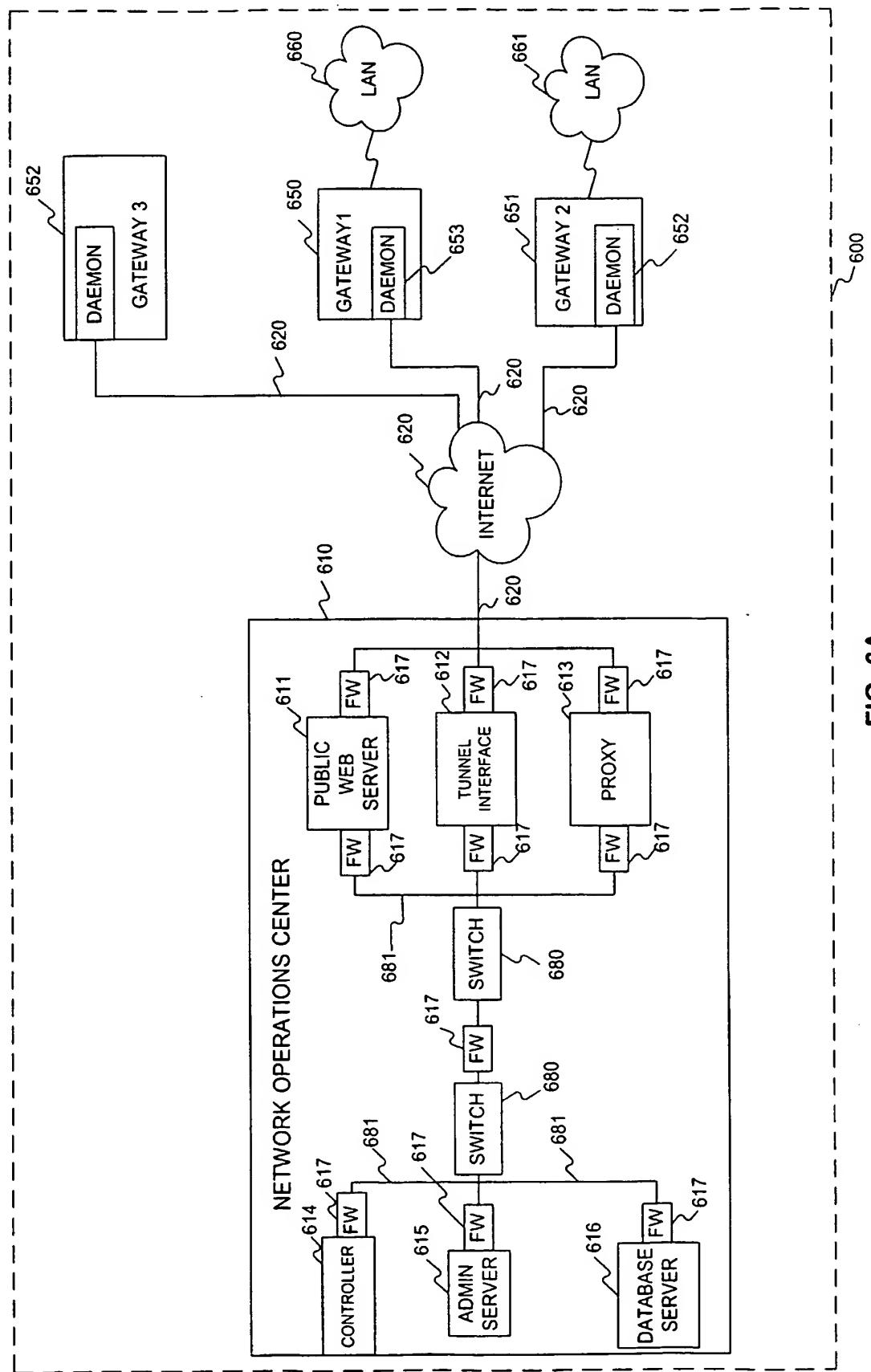
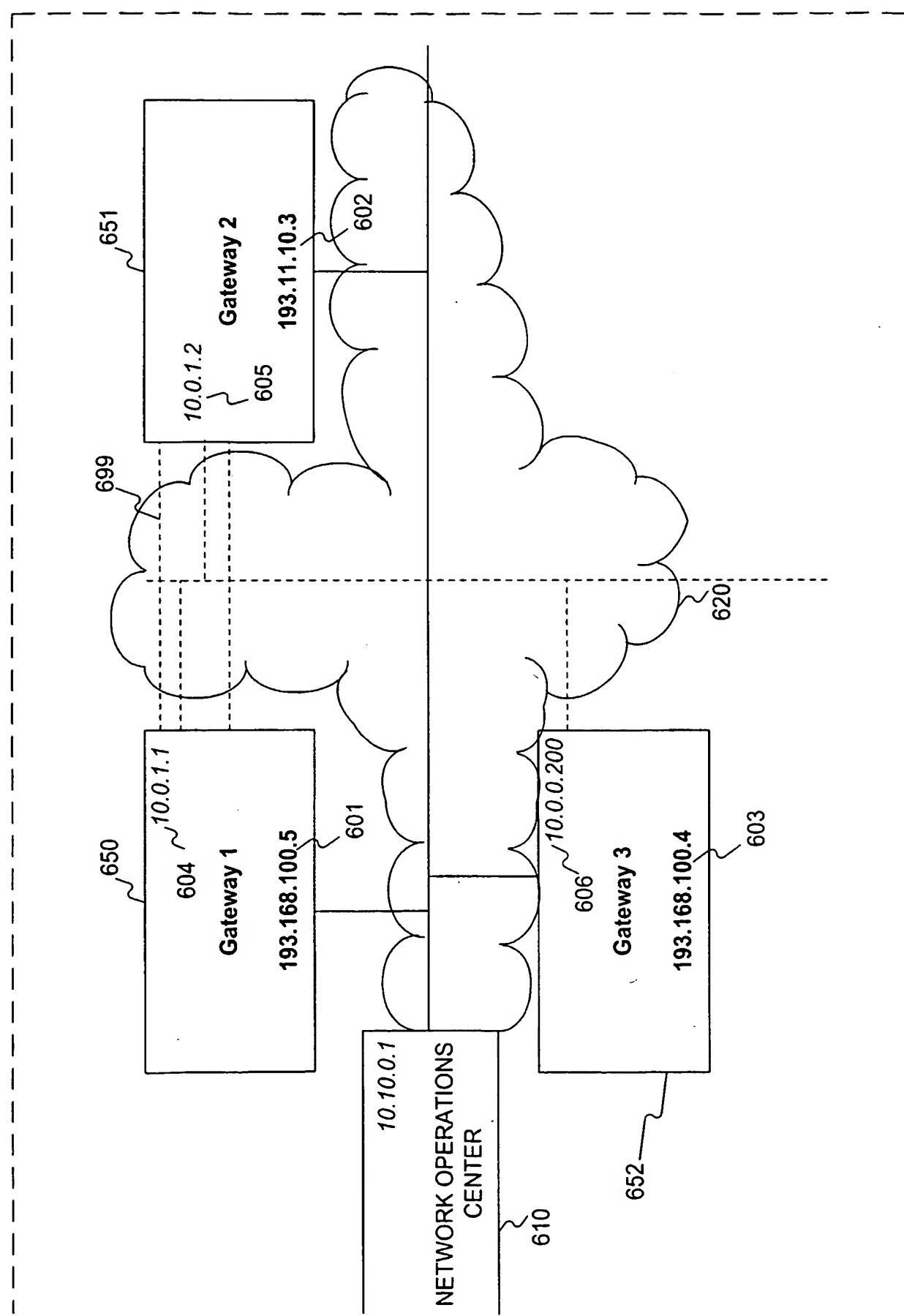


FIG. 6A

THIS PAGE BLANK (USPTO)

**FIG. 6B**

THIS PAGE BLANK (USPTO)

8/23

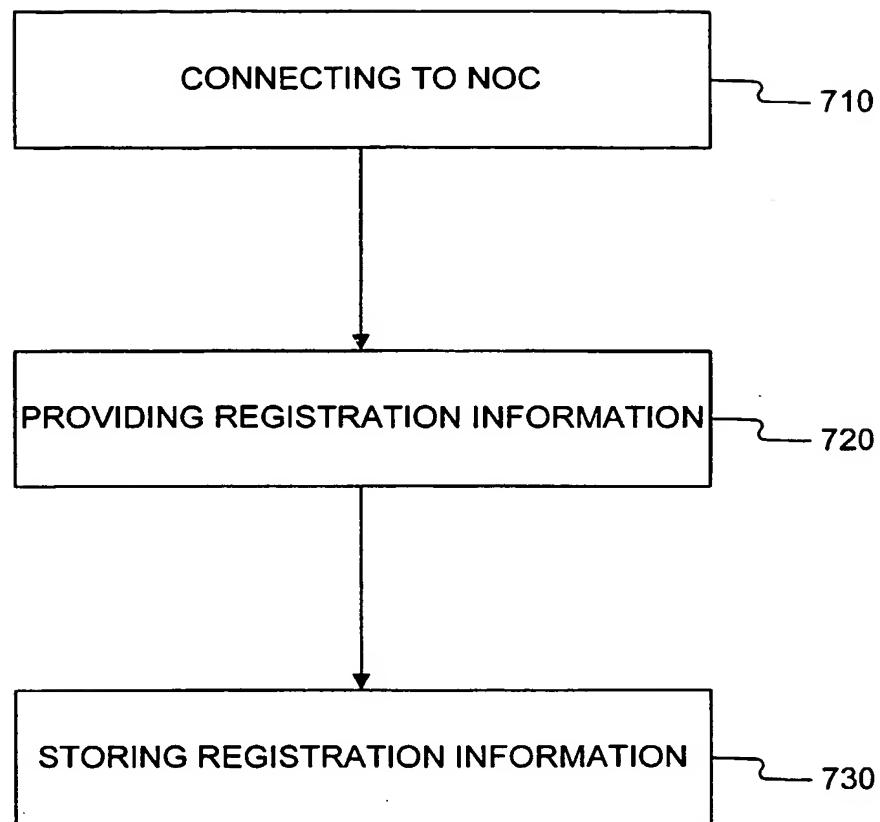
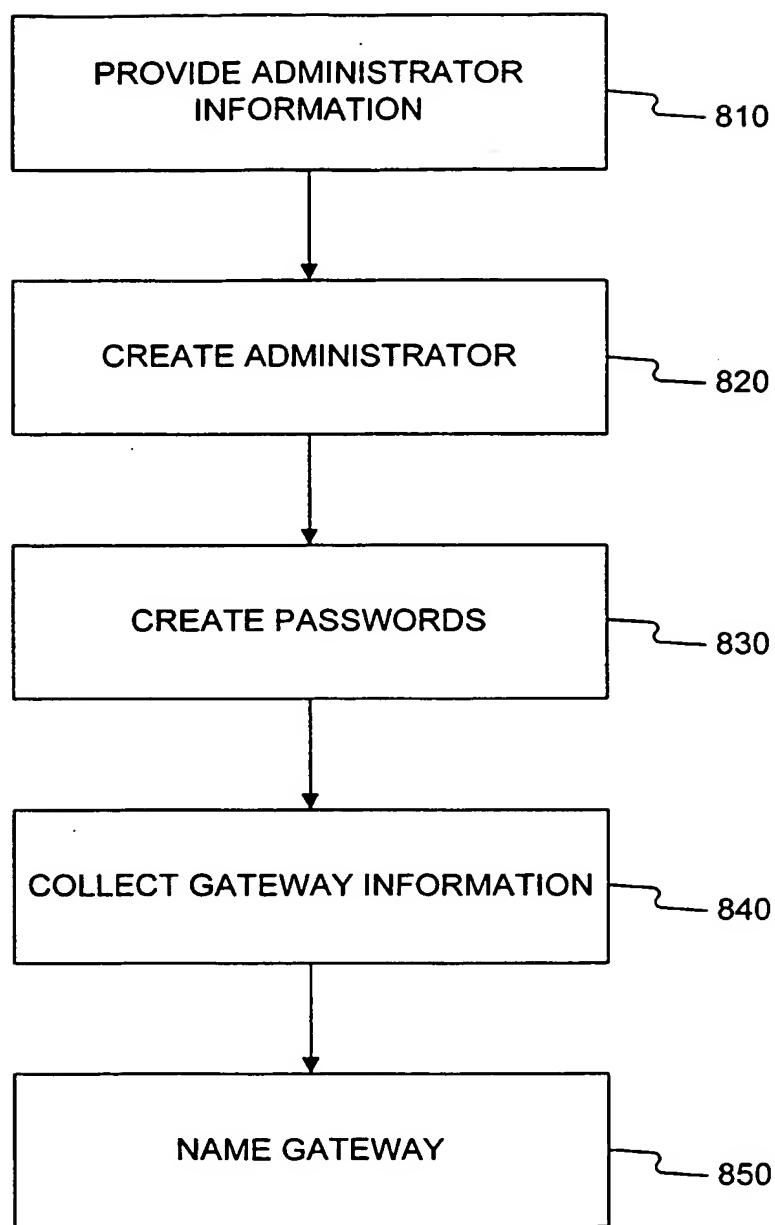


FIG. 7

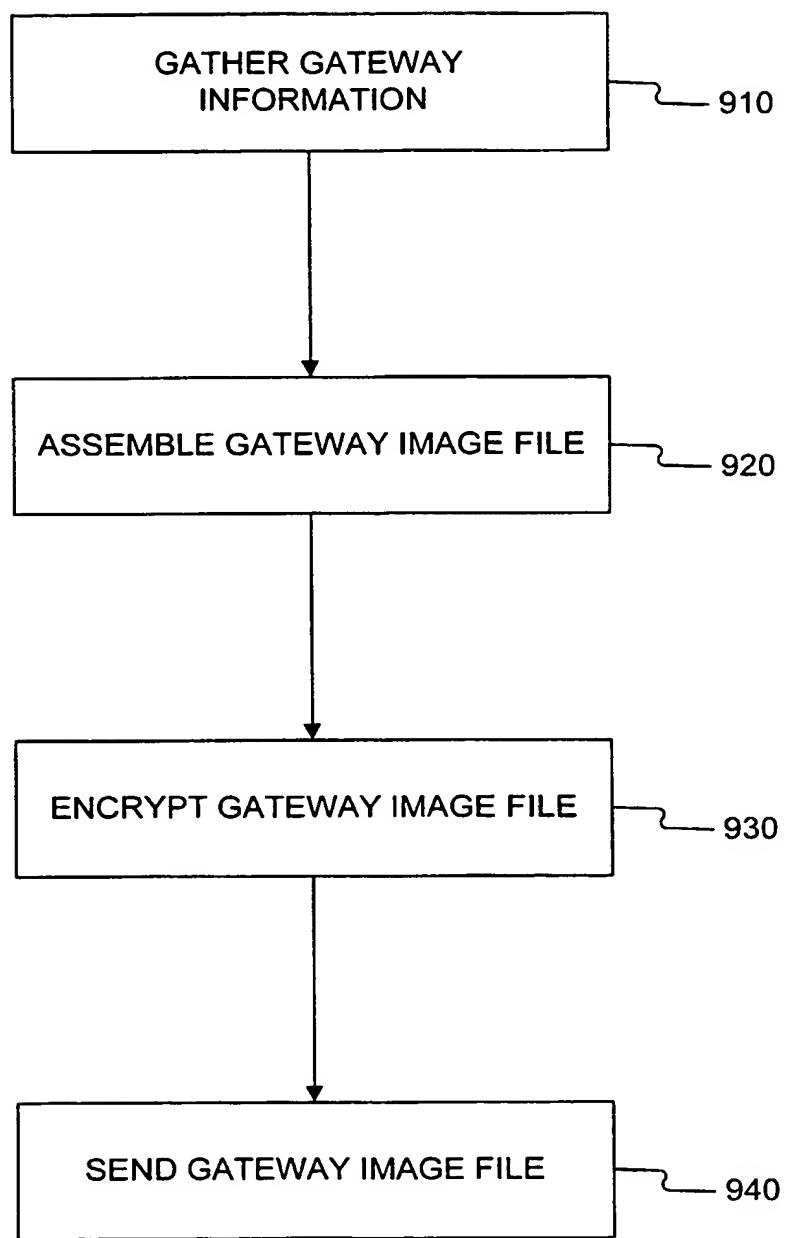
THIS PAGE BLANK (USPTO)

9/23

**FIG. 8**

THIS PAGE BLANK (USPTO)

10/23

**FIG. 9**

THIS PAGE BLANK (USPTO)

11/23

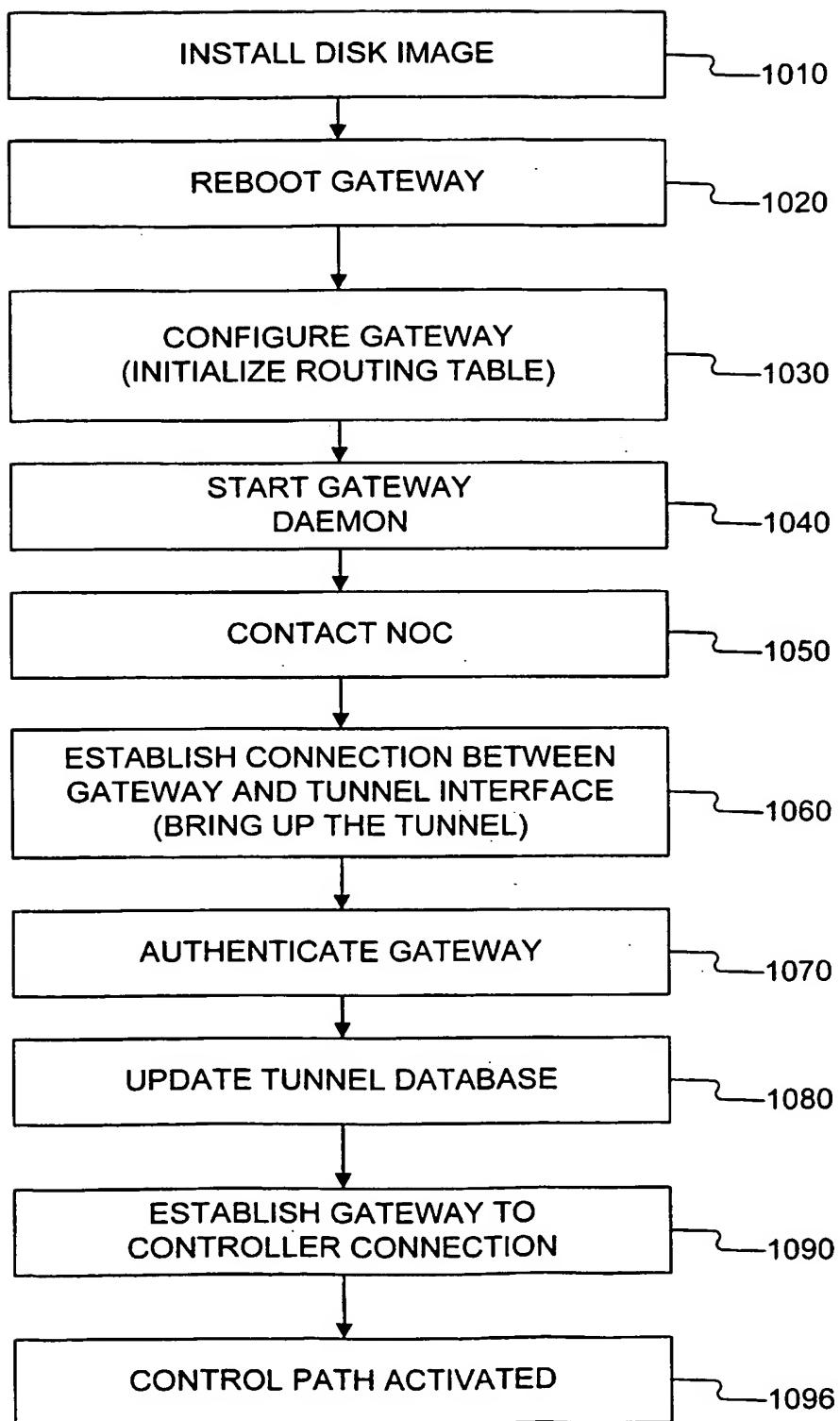
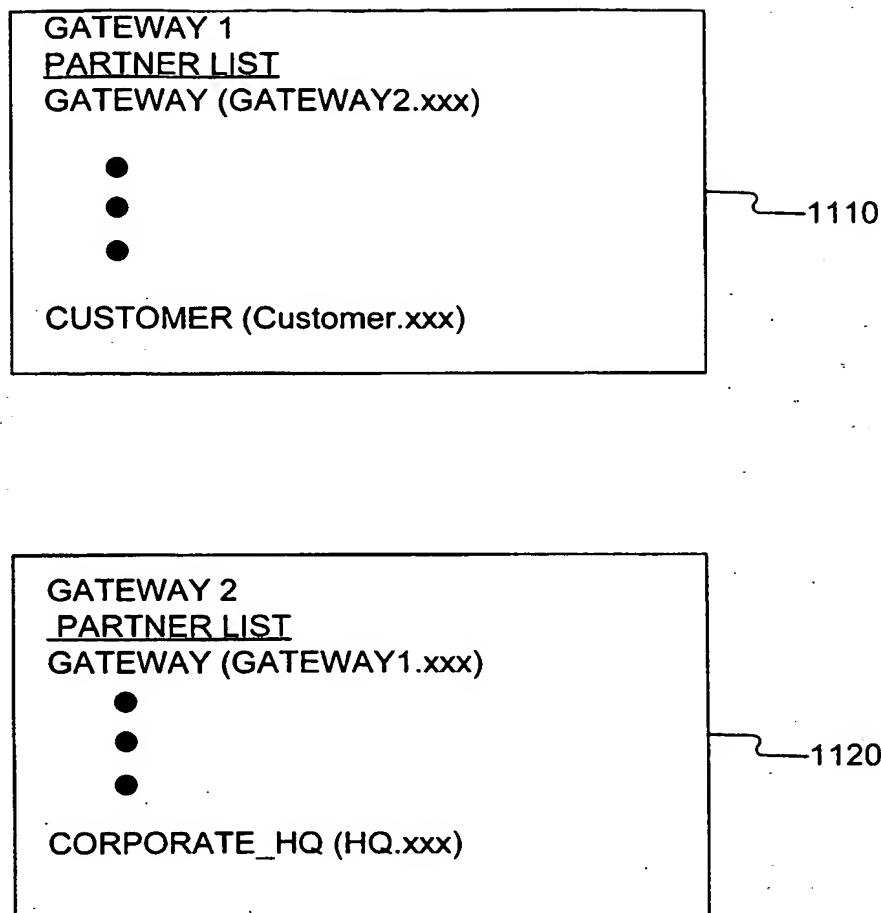


FIG. 10

THIS PAGE BLANK (USPTO)

12/23

**FIG. 11**

THIS PAGE BLANK (USPTO)

13/23

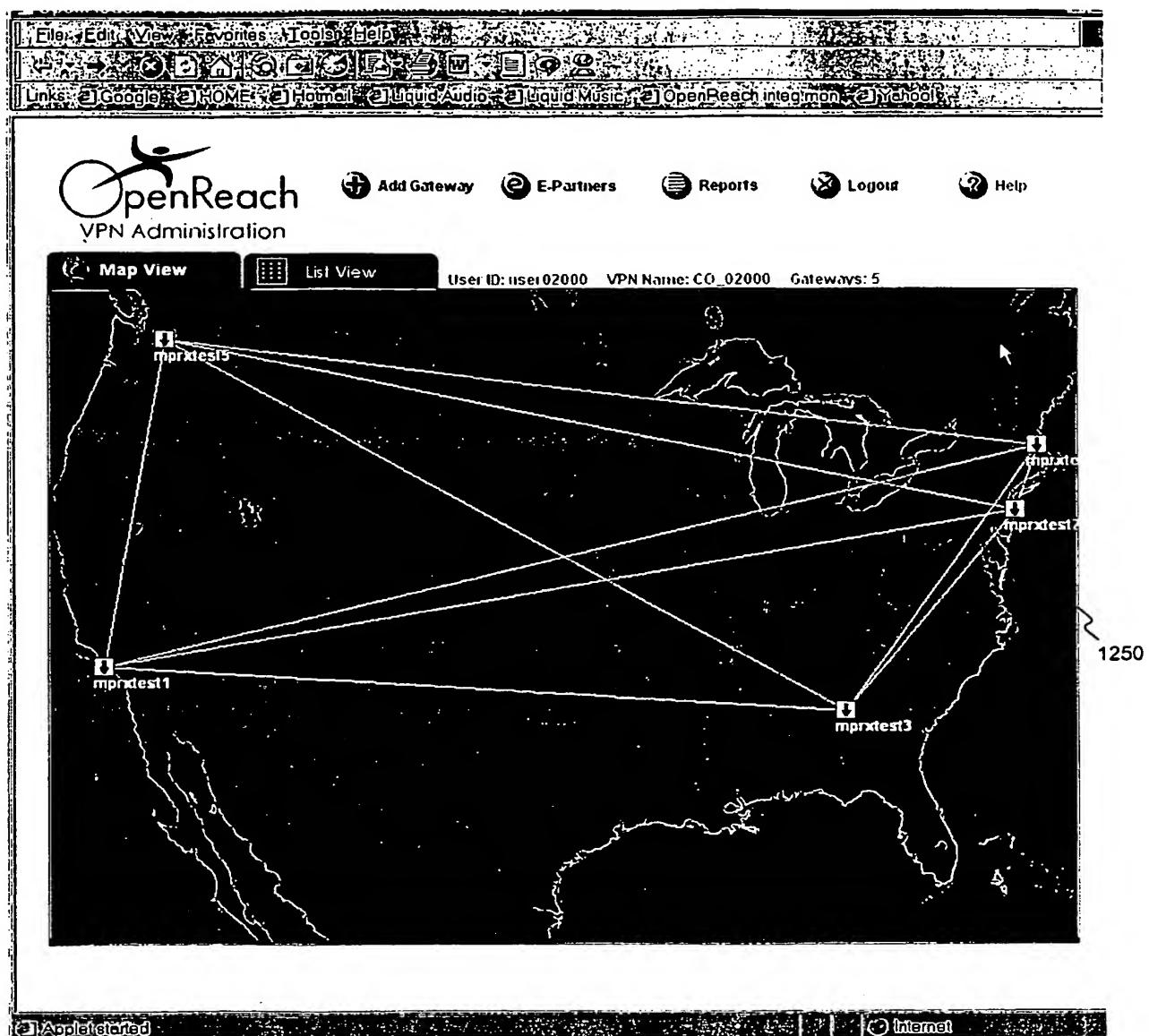


FIG. 12

THIS PAGE BLANK (USPTO)

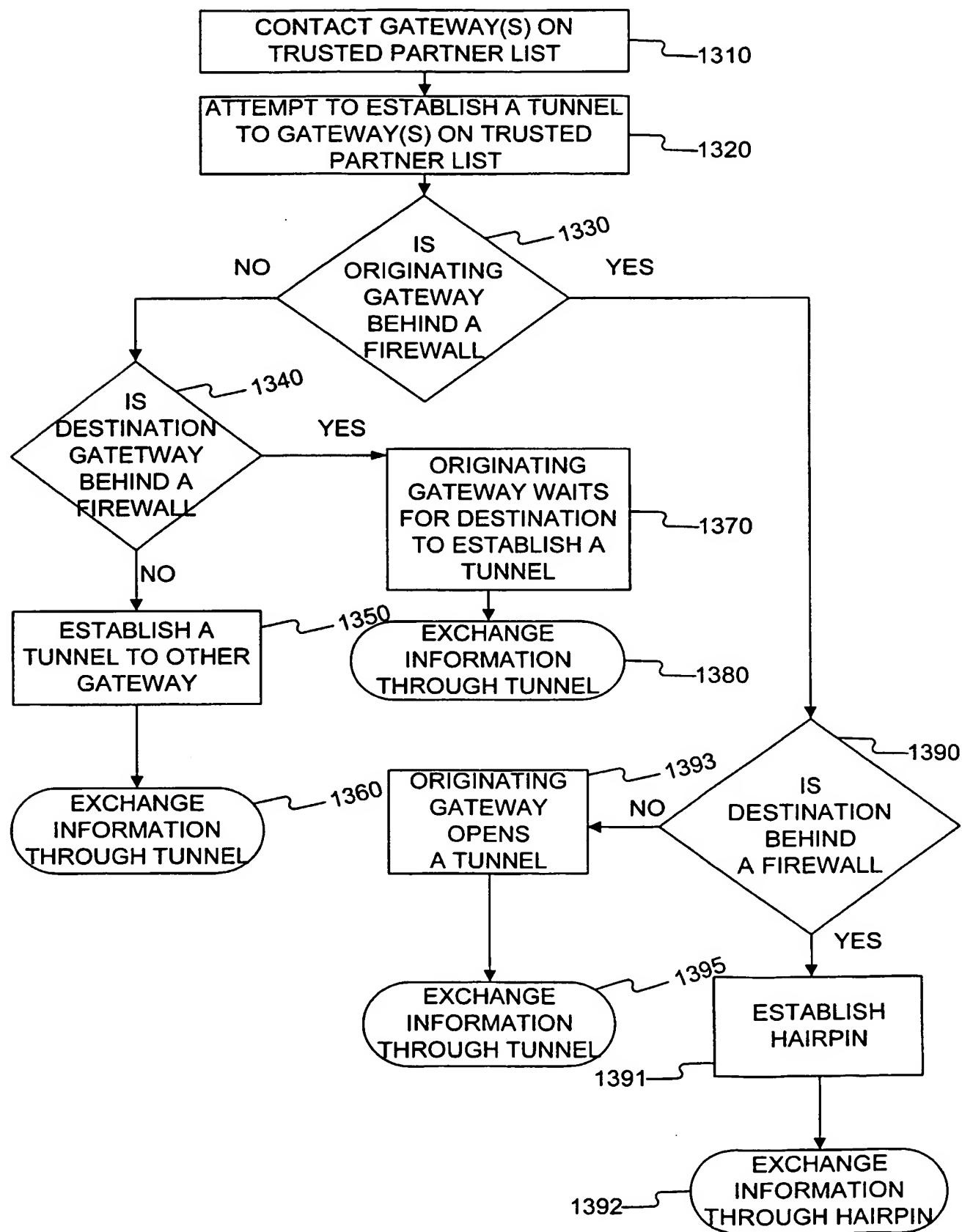
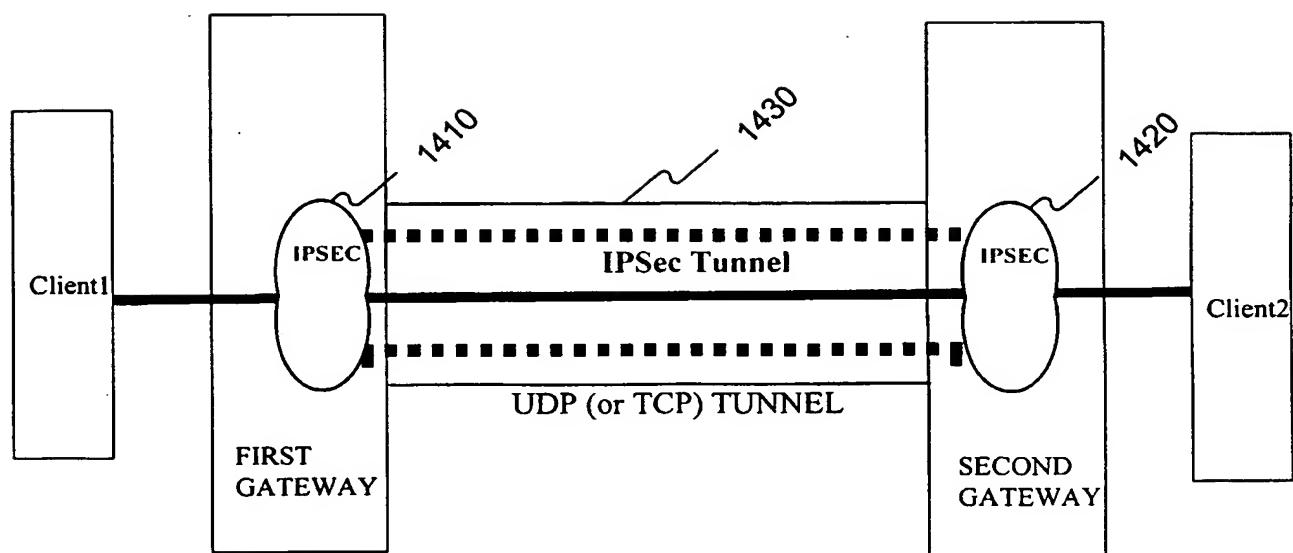
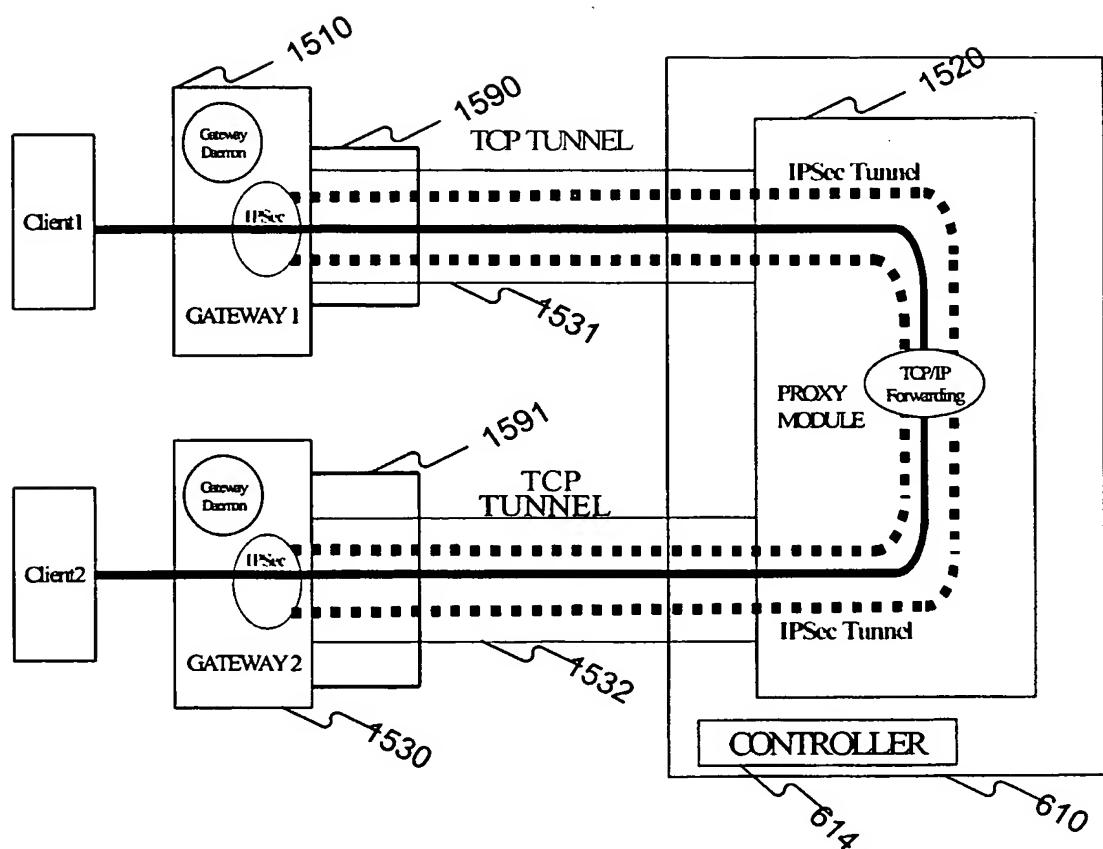


FIG. 13

THIS PAGE BLANK (USPTO)

**FIG. 14**

THIS PAGE BLANK (USPTO)

1500**FIG. 15**

THIS PAGE BLANK (USPTO)

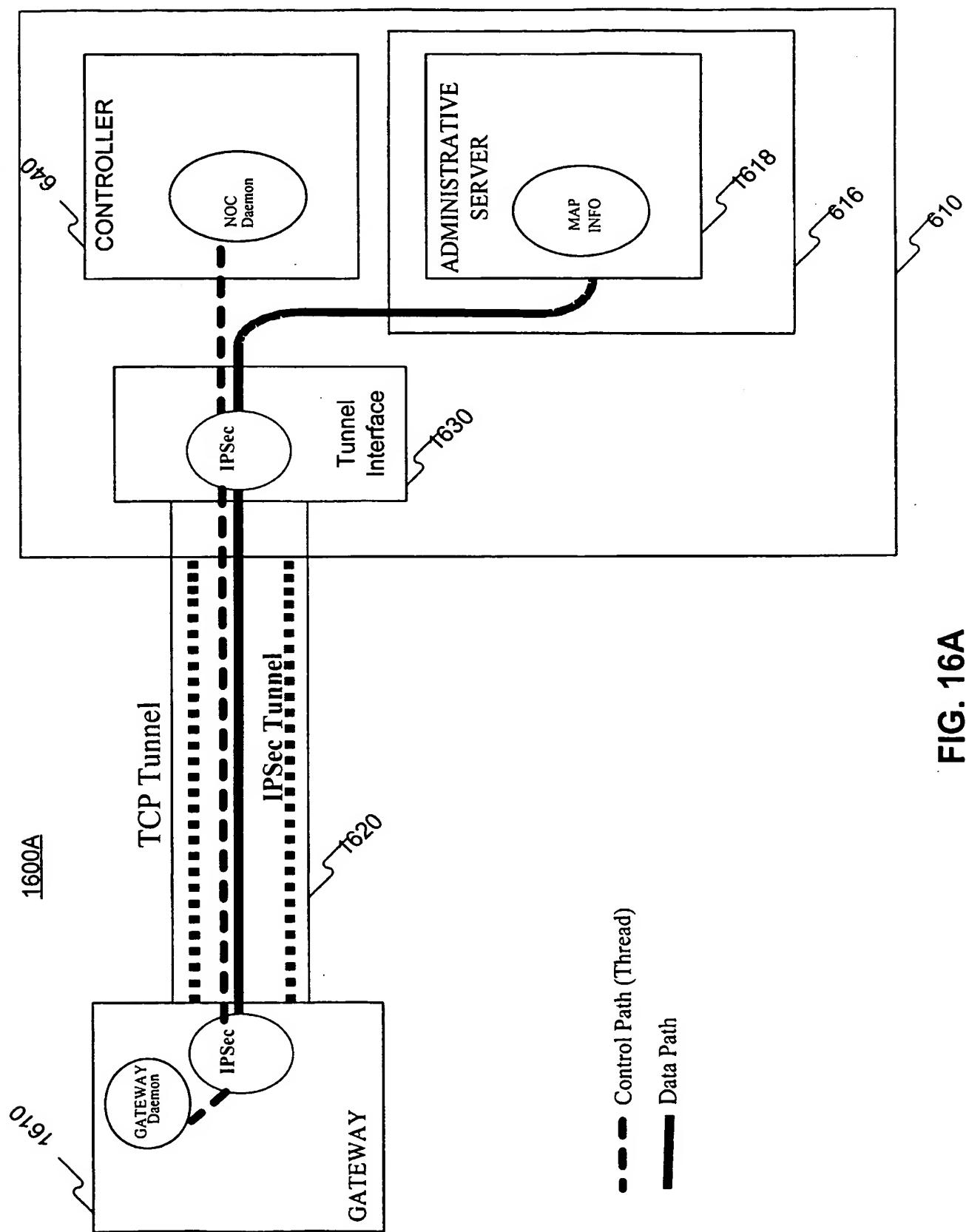
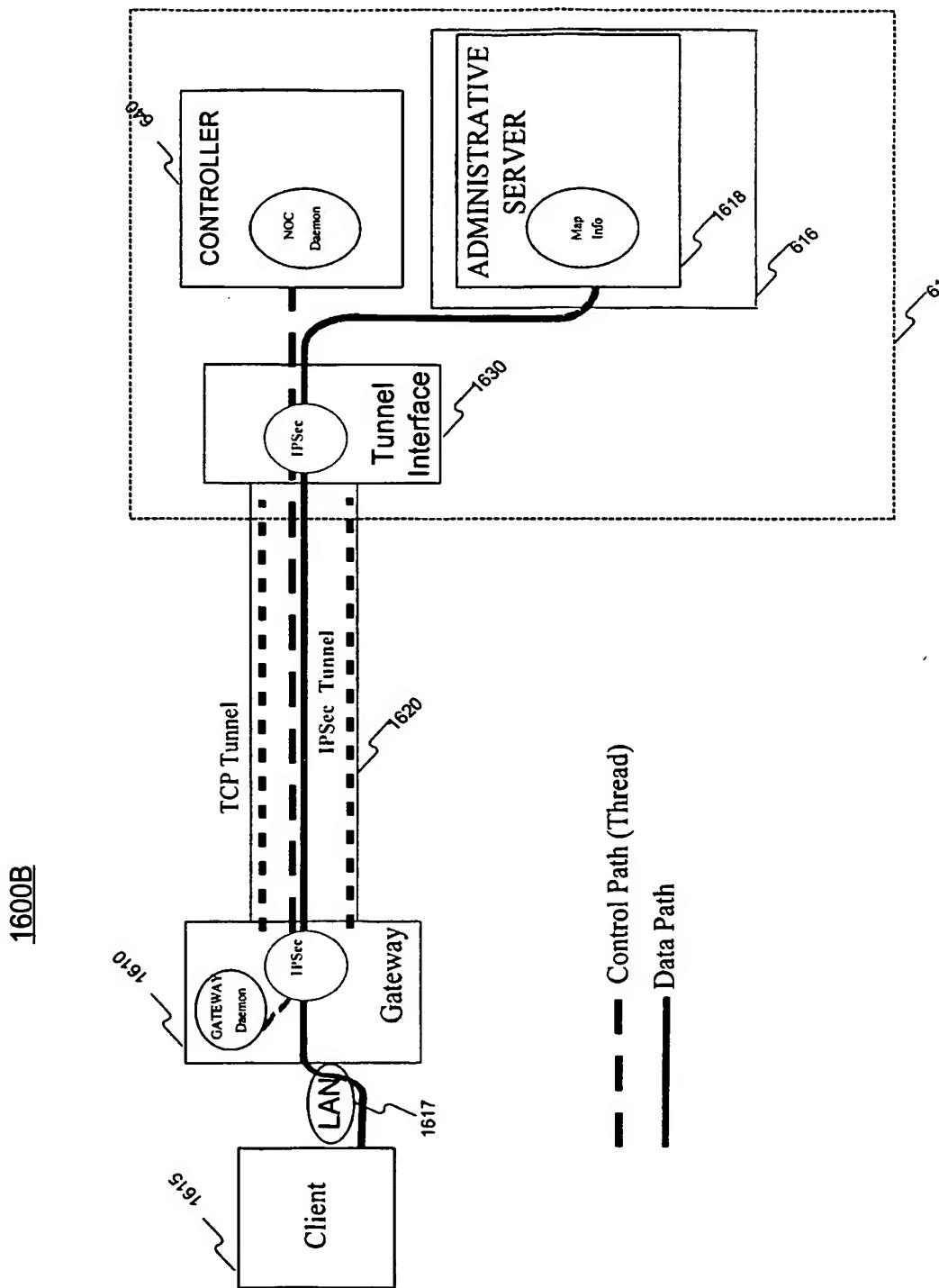


FIG. 16A

THIS PAGE BLANK (USPTO)

**FIG. 16B**

THIS PAGE BLANK (USPTO)

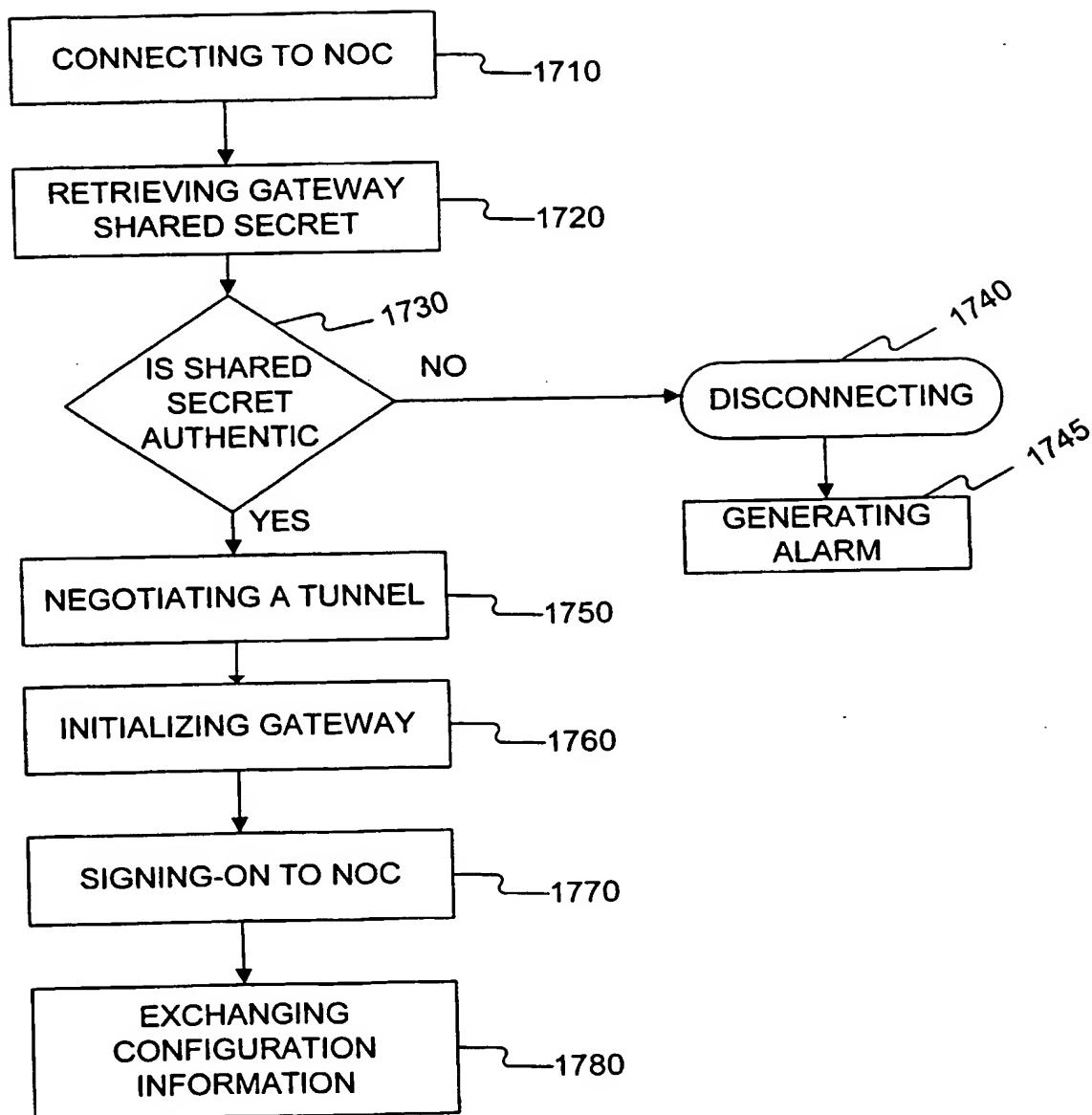
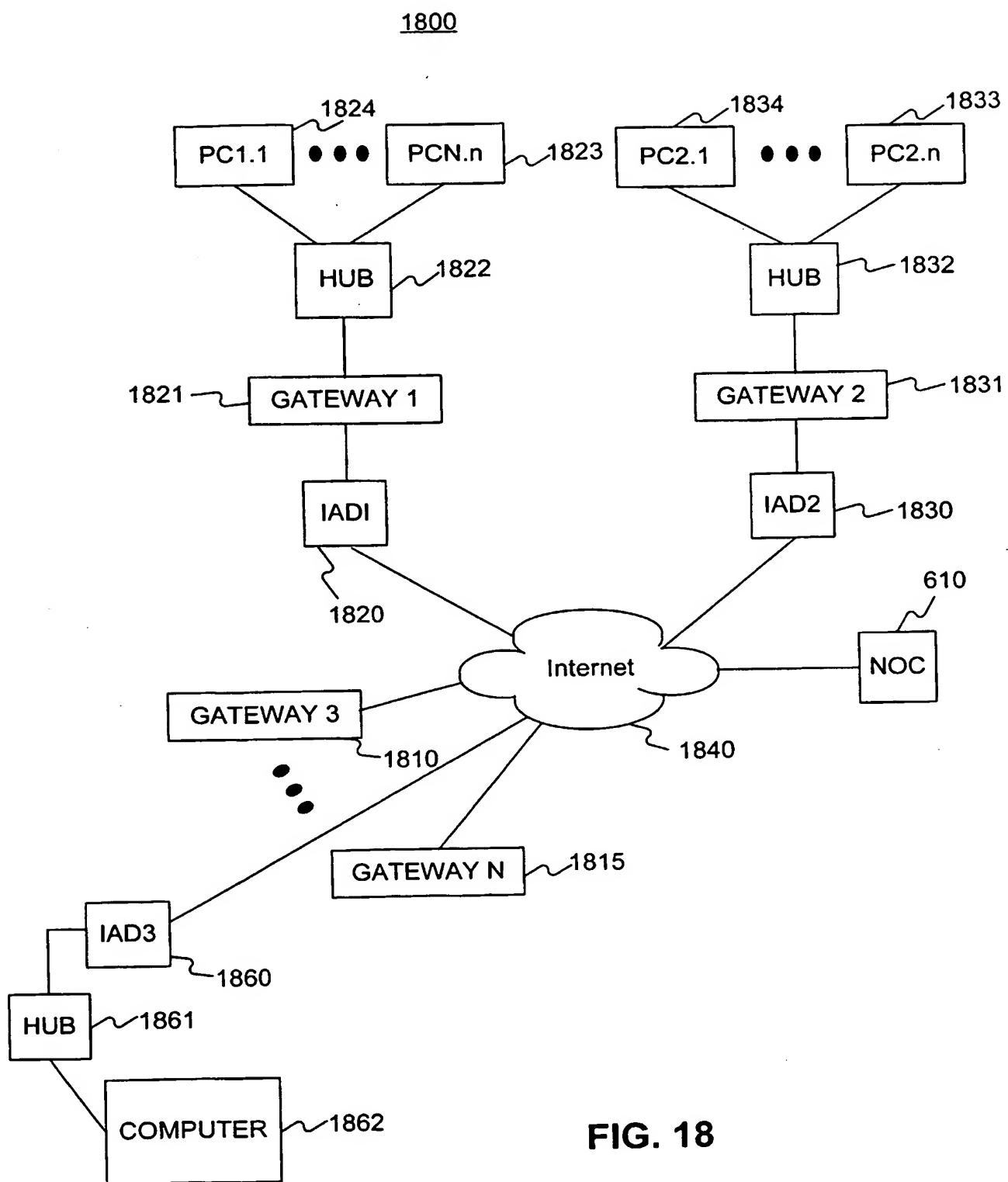


FIG. 17

THIS PAGE BLANK (USPTO)



THIS PAGE BLANK (USPTO)

21/23

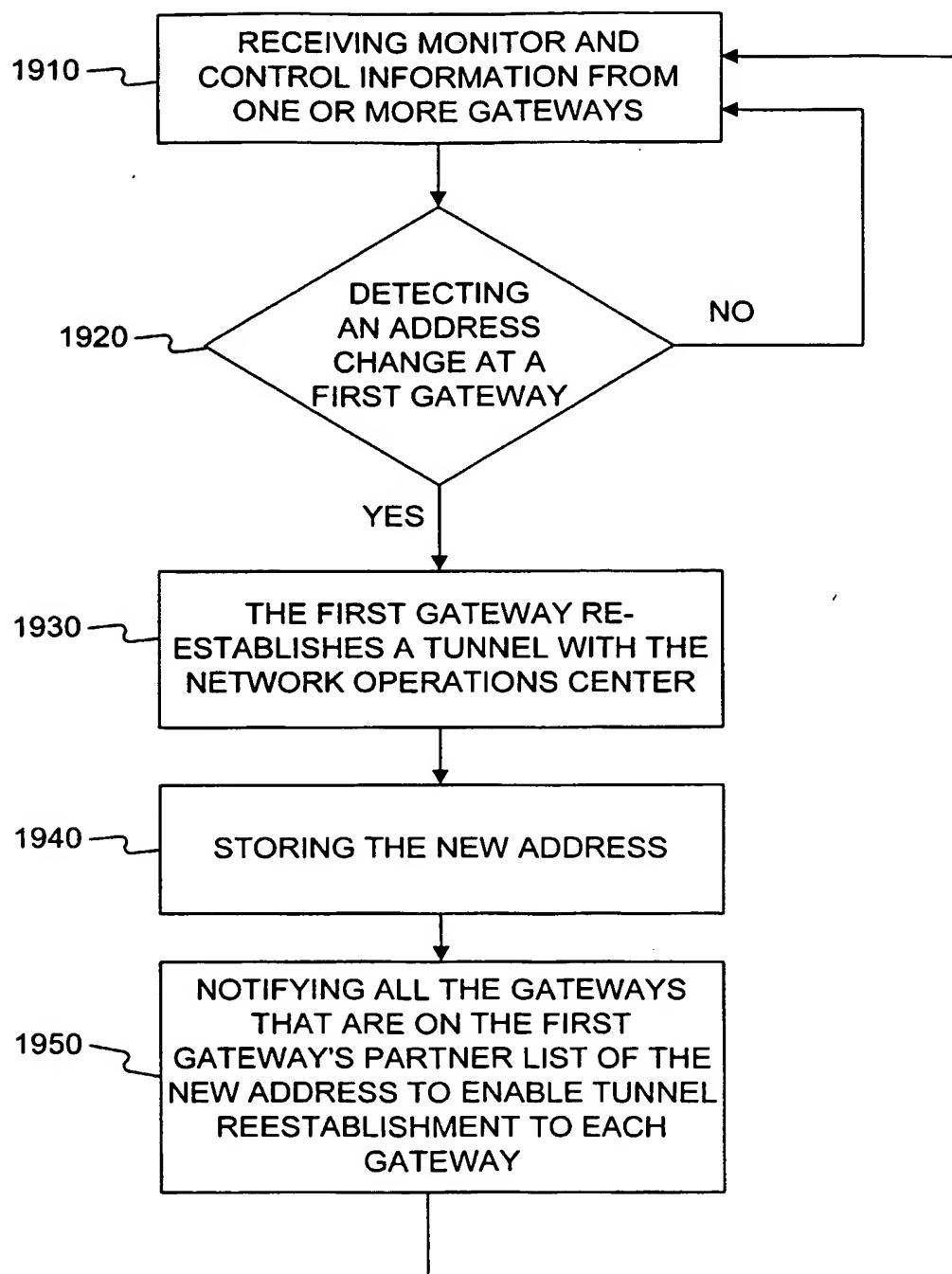


FIG. 19

THIS PAGE BLANK (USPTO)

22/23

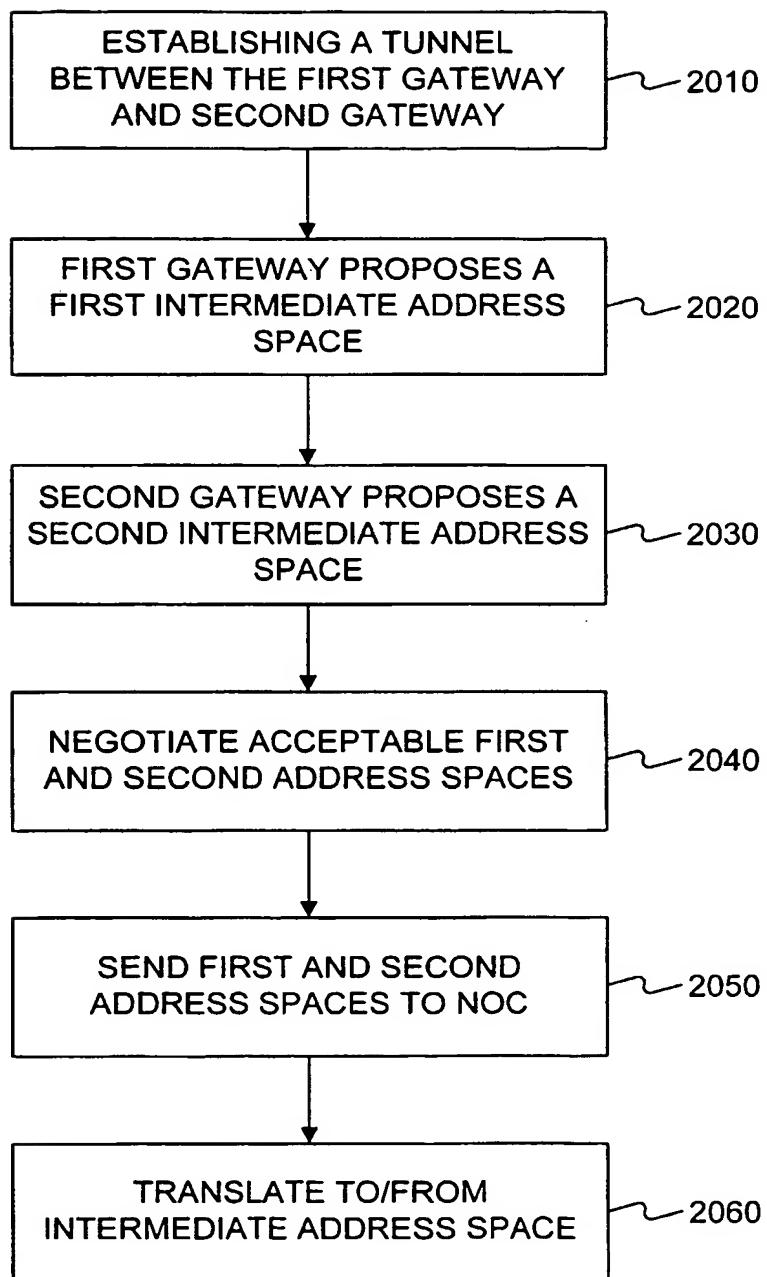


FIG. 20

THIS PAGE BLANK (USPTO)

23/23

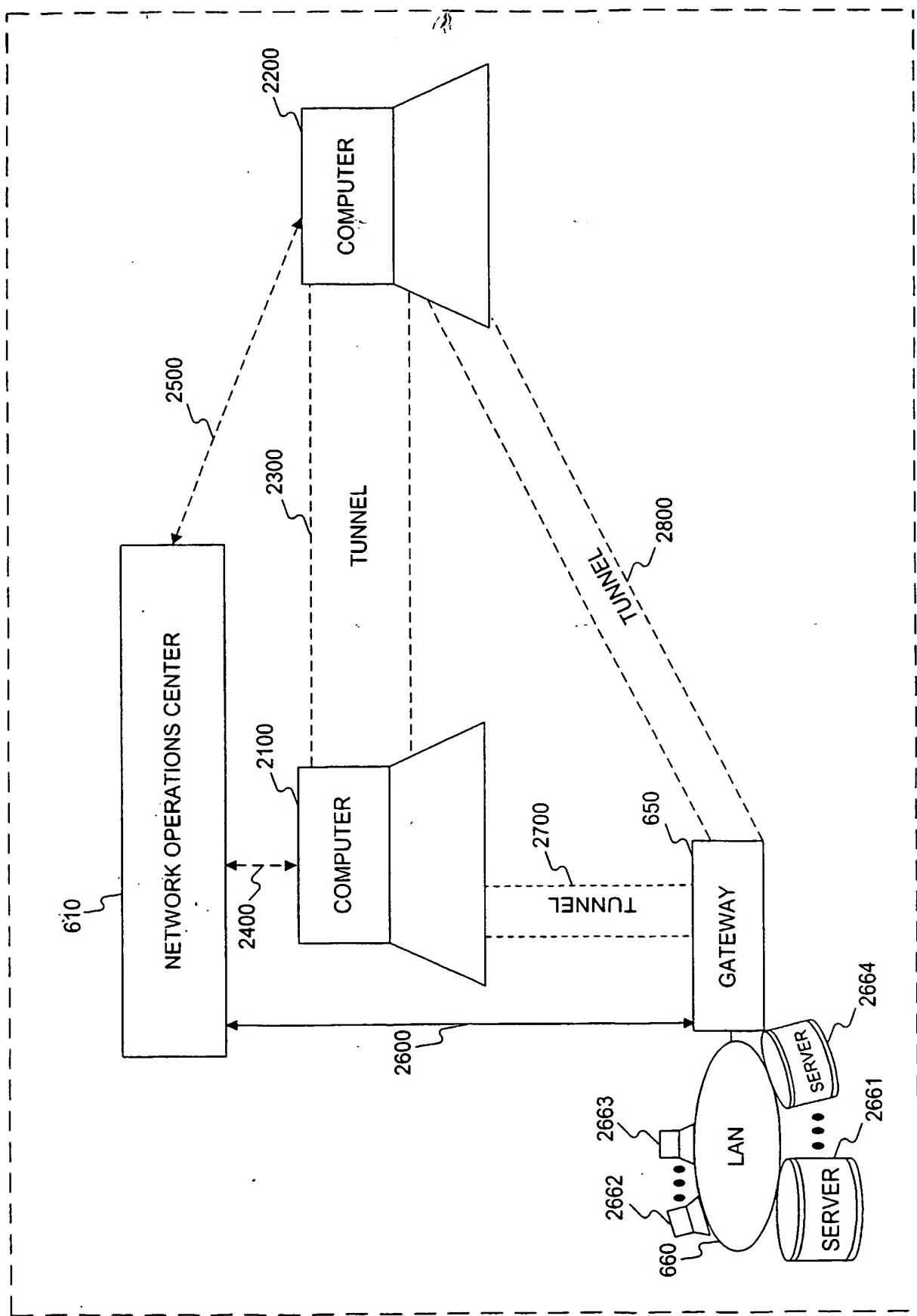


FIG. 21

THIS PAGE BLANK (USPTO)

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- BLACK BORDERS**
- IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- FADED TEXT OR DRAWING**
- BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- SKEWED/SLANTED IMAGES**
- COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- GRAY SCALE DOCUMENTS**
- LINES OR MARKS ON ORIGINAL DOCUMENT**
- REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- OTHER: _____**

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.

THIS PAGE BLANK (USPTO)